IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 3, Issue 8, August 2016

Effective in Latency and Complexity with Multiple Butterfly-Formed Weight Accumulators

G. Swapna[1], D. Shyam Prasad[2]

M.Tech (VLSI) [1]

Associate Professor [2]

Abstract: Data comparison is widely used in computing system to perform many operations. Where incoming information is needs to be compared with a piece of stored data to locate the matching entry. If both incoming bits and stored bits are matching means there is no error if mismatched means some type of error will occur like random error or burst error. To detect and correct the error here error correcting codes are used. To further reduce the latency and complexity, in addition, a new butterfly-formed weight accumulator (BWA) is proposed for the efficient computation of the Hamming distance. The proposed architecture examines whether the incoming data matches the stored data if a certain number of burst errors are corrected. The basic function of the BWA is to count the number of 1's among its input bits. It consists of multiple stages of HAs where each output bit of a HA is associated with a weight. The HAs in a stage are connected in a butterfly form so as to accumulate the carry bits and the sum bits of the upper stage separately.

Keywords: Error correcting codes, hamming distance, data comparison, parity matrix.

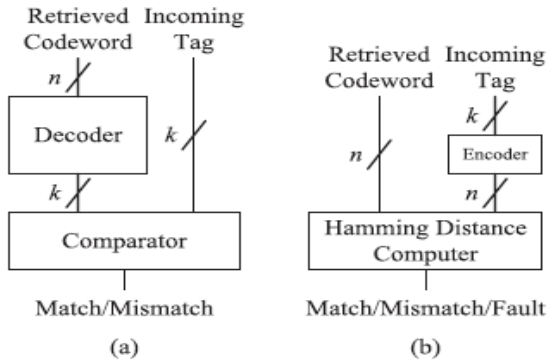
1. INTRODUCTION

Now a day's data comparison is widely used in much application. It also performs many operations. This comparison circuit consists of low hardware complexity. System performance is increased besides the data comparison usually resides in the critical path of the components that are devised to increase the system performance. Whose outputs determine the flow of the succeeding operations in a pipeline .Therefore the circuit must be designed to have as a low latency as possible overall performance of the whole system would be severely deteriorated. To protect the data error correcting code is used to improve the reliability ECC decoding and correction before the comparison operation can be performed. To eliminate the complex decoding from the critical path. This method does not represent the Whether the retrieved data is extractly matched with the incoming data. To calculate the hamming distance which represents the difference between two numbers. To separate the data part and parity part systematic form is used.

2. EXISTING METHODS

This section describes the two methods encode compare method and decode compare method .In first method retrieved bit is directly given to the hamming distance and incoming bit encoded after given to the hamming distance. This hamming distance determine the both bits are matched are mismatched or any fault in the bits. It takes more time to detect the error.

To overcome the above drawback other method is used. Decode and compare architecture here incoming data is directly given to the comparator and retrieved data given to decoder after performing decoding operation compactor check whether the bits are matched or mismatched. In this method complexity is increased and most complicated to process this method.



a) Decode and compare structure b) Encode and compare architecture

To calculate the hamming distance d is used to represent the range. Some conditions are given below.

1) If d = 0, X matches Y exactly.

2) If 0 < d ≤ tmax, X will match Y provided at most tmax errors in Y are corrected.

3) If tmax ☐☐d ≤ rmax, Y has detectable but uncorrectable errors.In this case, the cache may issue a system fault so as to make the central processing unit take a proper action.

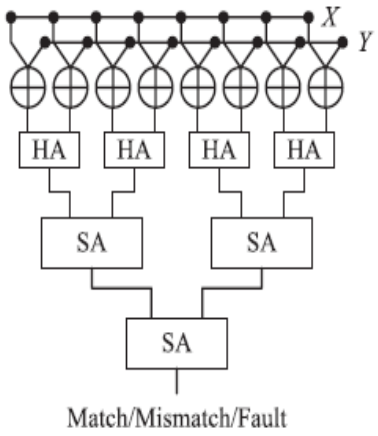
4) If rmax ☐☐d, X does not match Y.

DOI 10.17148/IARJSET.2016.3814

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 3, Issue 8, August 2016



3. PROPOSED ARCHITECTURE

In this section new architecture is proposed to reduce the delay and complexity of the data comparison by using systematic codes. Here incoming bit is encoded.ECC codeword is of systematic form which can separated completely. In this proposed structure after completing the encoding process to generate the parity bits used to reducing the parity bits.

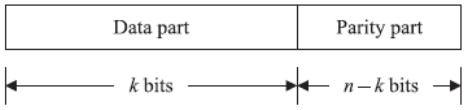


Fig.3 systematic representation of an ECC codeword

In this proposed structure consist of retrieved codeword n and incoming tag k both are combined and given to the XOR bank. Incoming tag is encoded after given to the XOR bank both are calculating the hamming distances. Finally decision unit is mainly used to decide the bits matched are mismatched. Here n-k represent the parity part. parity bit is generally called as a bit, with a value of 0 or 1,that is added to a block of data for error detection purposes. It gives the data either an odd or even parity, which is used to validate the integrity of the data parity bits are often used in data transmission to ensure that data is not corrupted during transfer process.

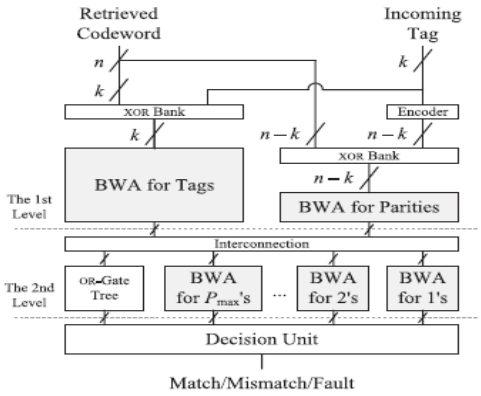


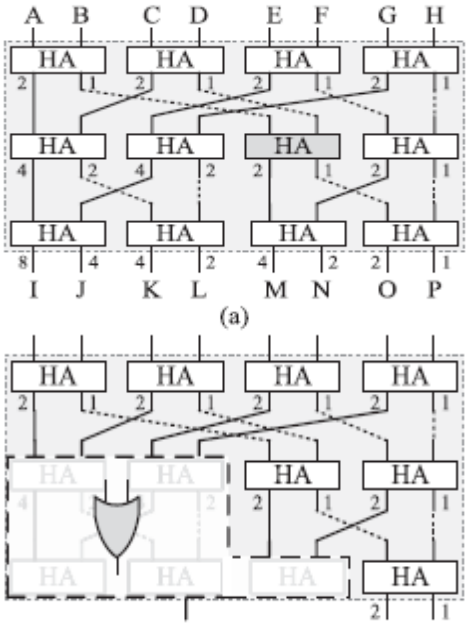
Fig 4. Proposed architecture optimized for systematic code words.

4. TO COMPUTING THE HAMMING DISTANCE

It contains multiple butterfly-formed weight accumulators proposed to improve the delay and complexity of the hamming distance computation. The function of BWA is to count the number of 1's among the input. It contains multiple stages of half adder each can associated with weight. Mainly it can be connected in the butterfly form to accumulate the carry bits and sum bits. The path reaching the half adder is equal to the weight of the output d can mainly calculated as,

$$D = 8I + 4 (J + K + M) + 2 (L + N + O) + P.$$

In fig 5(a) represents each half adder contains two input and associate with some weights. To reduce the number of half adders OR gate is used. Because to avoid the overlap between sum and carry bits It generates the bitwise difference vector for either data bits or parity bits, and the following processing elements count the number of 1's in the vector, the Hamming distance.



structure for matching the data. Each BWA at the first level is in the revised form and generates an output from the OR-gate tree and several weight bits from the HA trees. Let consider (8,4) single error correction and double error detection code.

TABLE I
TRUTH TABLE OF THE DECISION UNIT FOR A (8, 4) CODE

Q OR R OR S	T	U	V	Decision
	0	0	x	Match
	0	1	x	Fault
0	1	0	0	Fault
	1	0	1	Mismatch
	1	1	x	Mismatch
1	x	x	x	Mismatch

DOI 10.17148/IARJSET.2016.3814

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 3, Issue 8, August 2016

The Truth for the decision unit finally determines if the incoming tag is matches with the retrieved cord word consider four ranges using hamming distance.

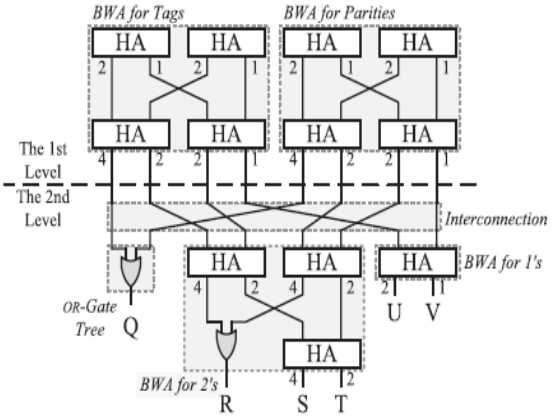


Fig.6. First and second level circuits for (8,4) code.

5. GENERAL EXPRESSION FOR DELAY AND COMPLEXITY

The algorithm mainly depends on the latency of combinational circuits the complexity and the latency are usually conflicting with each other.

Non deterministic algorithm is used to estimate the delay and complexity. The proposed structure complexity c represents,

C = CXOR+ CENC+ CBWA(k) + CBWA(n− k)+ C2nd + CDU ≤ n + CENC + 2CBWA(n) + CDU

Where CXOR, CENC, C2nd, CDU, and CBWA□n□ □are the complexities of XOR banks, an encoder, the second level circuits.

CBWA(n) can be calculated as
CBWA(n) = CBWA (n/2) + CBWA (n/2) + 2 [n/2]

Where the seed value, CBWA(1), is 0. Note that
When a + b =c, CBWA□a□ □+ CBWA□b□ □≤ CBWA□c□ □holds for all positive integers a, b□ □and c.

Because of the inequality and the fact that an OR-gate tree for n inputs is always simpler than a BWA for n inputs, both CBWA□k□ □+ CBWA□n –k□ □and C2nd are bounded by

CBWA□n□.The latency of the proposed architecture, L, can be expressed as,

L ≤ max [LXOR + LBWA(k), LENC + LXOR + LBWA(n − k)] +L2nd+LDU

Where LXOR, LENC, L2nd, LDU, and LBWA□n□ □are the latencies of an XOR bank, an encoder, the second level circuits, the decision uni

ECC	Architecture	Gate-Level Counting		Implementation Results	
		Latency[a]	Complexity[b]	CPD[c]	EGC[d]
(16, 11)	Conventional	14 (1.17)[e]	137 (1.10)	2.13 (1.16)	320 (1.20)
	SA-based	14 (1.17)	132 (1.06)	2.12 (1.16)	304 (1.14)
	Proposed	12 (1.00)	125 (1.00)	1.83 (1.00)	266 (1.00)
(24, 18)	Conventional	16 (1.23)	238 (1.24)	2.31 (1.16)	491 (1.19)
	SA-based	18 (1.38)	211 (1.10)	2.46 (1.23)	475 (1.15)
	Proposed	13 (1.00)	192 (1.00)	2.00 (1.00)	412 (1.00)
(31, 25)	Conventional	16 (1.23)	336 (1.29)	2.48 (1.24)	684 (1.22)
	SA-based	18 (1.38)	290 (1.11)	2.57 (1.29)	634 (1.13)
	Proposed	13 (1.00)	261 (1.00)	1.99 (1.00)	561 (1.00)
(40, 33)	Conventional	18 (1.20)	473 (1.38)	2.64 (1.20)	861 (1.21)
	SA-based	22 (1.47)	377 (1.10)	2.96 (1.35)	816 (1.15)
	Proposed	15 (1.00)	342 (1.00)	2.20 (1.00)	709 (1.00)

[a]The number of gates in the critical path.
[b]The count of all the gates.
[c]The critical-path delay (CPD) in nanoseconds.
[d]The equivalent gate count (EGC) measured by counting a two-input NAND as one.
[e]The numbers in parentheses are normalized values.

The proposed architecture is effective in reducing the latency as well as the hardware complexity even with considering the practical factors. The latencies of the SA-based architecture and the proposed one are dominated by SAs and HAs, respectively. As the bit width doubles, at least one more stage of SAs or Has needs to be added Since the critical path of a HA consists of only one gate while that of a SA has several gates, the proposed architecture achieves lower latency than its SA-based counterpart, especially for long code words.

6. ERROR CORRECTION CODES

Error detection and correction (EDAC) techniques are used to ensure that data is correct and has not been corrupted, either by hardware failures or by noise occurring during transmission or a data read operation from Memory. There are many different error correction codes in existence. The reason for the different codes being used in different applications has to do with the historical development of the data storage, the types of data errors occurring, and the overhead associated with each of the error detection techniques.
The basic concept of error detection and correction method is as follow

1.Networks must be able to transfer data from one system to another without data can be corrupted during transmission.
2.For reliable communication, errors must be detected and corrected. Any error-correcting code can be used for error detection and correction. Error-correcting code (ECC) or forward error correction (FEC) code is a system of adding redundant data, or parity data, to a message, such that it can be recovered by a receiver even when a number of errors were introduced, either during the process of transmission, or on storage.

Since the receiver does not have to ask the sender for retransmission of the data, a back-channel is not required in forward error correction, and it is therefore suitable for simplex communication such as broadcasting. Error-correcting codes are frequently used in lower-layer communication, as well as for reliable storage in media such as CDs, DVDs, hard disks, and RAM.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 3, Issue 8, August 2016

The general idea for achieving error detection and correction is to add some redundancy to a message, which receivers can use to check consistency of the delivered message, and to recover data determined to be corrupted. . Error-detection and correction schemes can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits. which arederived from the data bits by some deterministic algorithm. If only error detection is required.

7. CONCLUSION

A new architecture has been presented for matching the data protected with an ECC to reduce the complexity and delay. The proposed architecture examines whether the incoming data matches the tored data if a certain number of erroneous bits are corrected. Systematic codeword is used to enabled the based on parallel operation it has separate the data and parity bits. To reduce the latency, the comparison of the data is parallelized with the encoding process that generates the parity information. an efficient processing architecture has been presented to further minimize the latency and complexity. The experimental results show that the efficiency of the proposed method. As the proposed architecture is effective in reducing the latency as well as the complexity considerably, it can be regarded as a promising solution for the comparison of ECC protected data.

REFERENCES

[1] 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 EncodedWith Hard Systematic Error-Correcting Codes" IEEE transactions on (vlsi) systems, vol. 22, no. 7, July 2014

[2] J. D. Warnock, Y.-H. Chan, S. M. Carey, H. Wen, P. J. Meaney, G. Gerwig and W. V. Huott "Circuit and physical design implementation of the microprocessor chip for the Enterprise system," IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 151–163, Jan. 2012.

[3] 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.

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

[5] Y. Lee, H. Yoo, and I.-C. Park, "6.4Gb/s multithreaded BCH encoder and decoder for multi-channel SSD controllers," in ISSCC Dig. Tech. Papers, 2012, pp. 426–427

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

[7] 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.