IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834, p- ISSN: 2278-8735.
PP 59-64
www.iosrjournals.org

Direct Comparison of Data Encoded With Hard Systematic Error Correcting Codes with Low Latency and Low Complexity

[1]Anju Thampi, [2]Meera Thampy
[1,2]Electronics and Communication Engineering Department,
Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala, India

***Abstract**: In current scenario, there are situations in a computing system where incoming information needs to be compared with a piece of stored data to locate the matching entry, e.g., cache tag array lookup and translation look-aside buffer matching. In this paper, a new architecture to reduce complexity and latency for matching the data protected with an error-correcting code (ECC). It is based on the fact that the codeword of an ECC generated by encoding is usually represented in a systematic form, and it consists of the raw data and the parity information. The proposed architecture parallelizes the comparison of the data and that of the parity information. To further reduce the latency and complexity, in addition, a modified 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 erroneous bits are correct.*

***Keywords**: Error correcting code (ECC), Butterfly Formed Weight accumulator (BWA), Data comparison.*

I. Introduction

Data comparison circuit is a logic that has many applications in a computing system. For example, to check whether a piece of information is in a cache, the address of the information in the memory is compared to all cache tags in the same set that might contain that address. Another place that uses a data comparison circuit is in the translation look-aside buffer (TLB) unit. TLB is used to speed up virtual to physical address translation. Because of such prevalence, it is important to implement the comparison circuit with low hardware complexity. Besides, the data comparison usually resides in the critical path of the components that are devised to increase the system performance, e.g., caches and TLBs, whose outputs determine the flow of the succeeding operations in a pipeline. 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.

Error correcting codes (ECCs) are widely used in modern microprocessors [7] to enhance the reliability and data integrity of their memory structures. For example, caches on modern microprocessors are protected by ECC. If a memory structure is protected with ECC, a piece of data is encoded first and the entire codeword including the ECC check bits are written into the memory array. When the input data is loaded into the system, it has to be encoded and compared with the data stored in the memory and corrected if errors are detected to obtain the original data.

This paper is organized as follows: Section II is a previous work. Proposed method is given in Section III. Evaluation is presented in Section IV. Conclusion and future work is given in Section IV.

II. PREVIOUS WORKS

A. Decode-And-Compare Architecture:

This section describes the conventional decode-and-compare architecture based on the direct compare method. Let us consider a cache memory where a k-bit tag is stored in the form of an n-bit codeword after being encoded by a (n, k) code. In the decode-and compare architecture, the n-bit retrieved codeword should first be decoded to extract the original k-bit tag. The extracted k-bit tag is then compared with the k-bit tag field of an incoming address to determine whether the tags are matched or not. As the retrieved codeword should go through the decoder before being compared with the incoming tag, the critical path is too long to be employed in a practical cache system designed for high speed access. Figure (1) is shown in below.

B. Direct Compare Method*:*

Direct compare method [3] is one of the most recent solutions for the matching problem. The direct compare method 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 the direct compare method,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. As the checking necessitates an additional circuit to compute the Hamming distance use a saturating adder. SA-based approach is the one where a special counter is constructed with an additional` building block called saturating adder (SA).The SA-based direct compare architecture [3] reduces the latency and hardware complexity by resolving the aforementioned drawbacks.

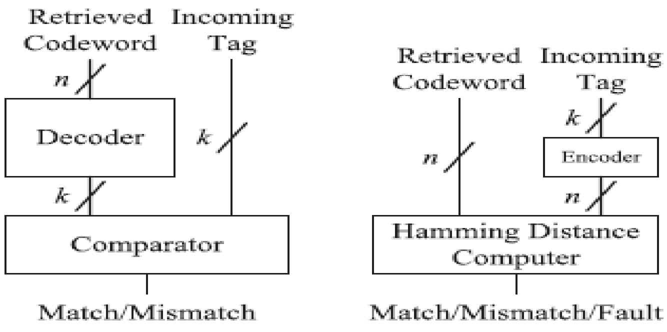


Fig: (1) Decode And Compare Method, (2) Direct Compare Method

III. PROPOSED ARCHITECTURE

This section presents a new architecture that can reduce the latency and complexity of the data comparison by using the characteristics of systematic codes.In this proposed structure consist of retrieved codeword and incoming tag are combined and given to the xor bank. Xor bank represents the array of bit-wise comparators (exclusive OR gates). It performs XOR operations for every pair of bits in X and Y so as to generate a vector representing the bitwise difference of the two codewords. The output from the XOR bank is then fed into BWA consisting of half adders (HAs). The numbers of 1's are accumulated by passing the value through the BWA.

The proposed architecture grounded on the data path design is given below. It contains multiple butterfly formed weight accumulators (BWAs) proposed to improve the latency and complexity of the Hamming distance computation. The basic function of the BWA is to count the number of 1's among its input bits.

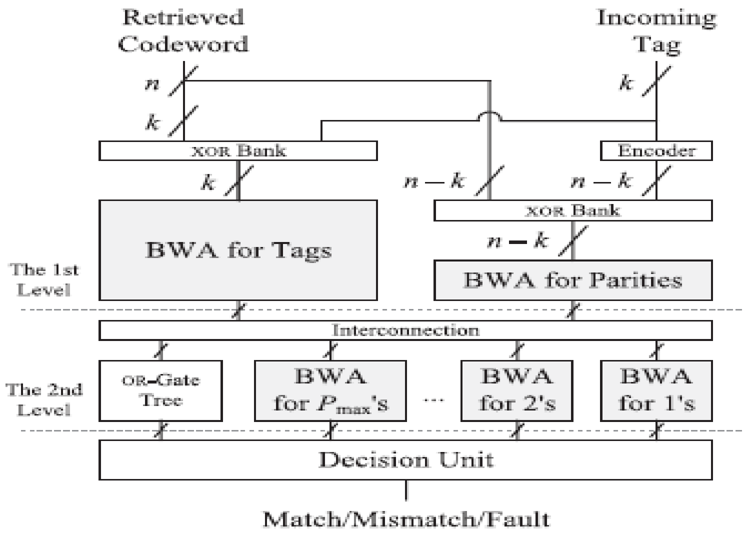


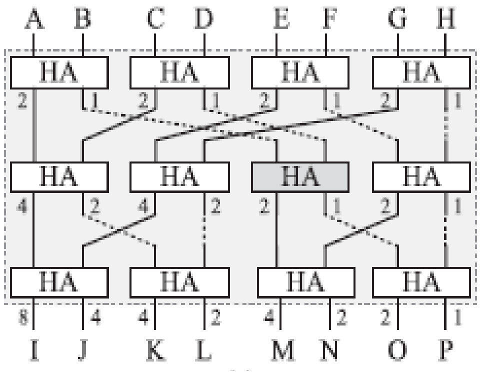
Fig: 3 Proposed data path design

The proposed architecture consists of multiple stages of HAs as shown in figure 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. In other words, both inputs of a HA in a stage, except the first stage, are either carry bits or sum bits computed in the upper stage. This connection method leads to a property that if an output bit of a HA is set, the number of 1's among the bits in the paths reaching the HA is equal to the weight of the output bit.

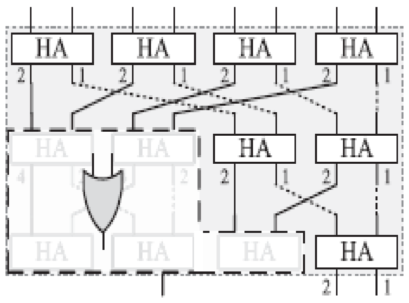
In the below figure for example, if the carry bit of the gray-colored HA is set, the number of 1's among the associated input bits, i.e., A, B, C, and D, is 2. At the last stage of above figure the number of 1's among the input bits, d, can be calculated as eq. (1)

$$d = 8\mathrm{I} + 4(\mathrm{J} + \mathrm{K} + \mathrm{M}) + 2(\mathrm{L} + \mathrm{N} + \mathrm{O}) + \mathrm{P}. \quad \text{eq(1)}$$

Since what we need is not the precise Hamming distance but the range it belongs to, it is possible to simplify the circuit. When r max = 1, for example, two or more than two 1's among the input bits can be regarded as the same case that falls in the fourth range. In that case, we can replace several HAs with a simple OR-gate tree as shown below. This is an advantage over the SA that resorts to the compulsory saturation.



(a)



(b)

*Fig: **4** (a) General structure of BWA & (b) Revised structure with OR-gate tree.*

Each XOR stage 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, i.e., the Hamming distance. Each BWA at the first level is in the revised form shown in figure above, and generates an output from the OR-gate tree and several weight bits from the HA trees. In the interconnection, such outputs are fed into their associated processing elements at the second level.

The output of the OR-gate tree is connected to the subsequent OR-gate tree at the second level, and the remaining weight bits are connected to the second level BWAs according to their weights. A simple (8, 4) single-error correction double-error detection code is considered and the corresponding first and second level circuits are shown below.

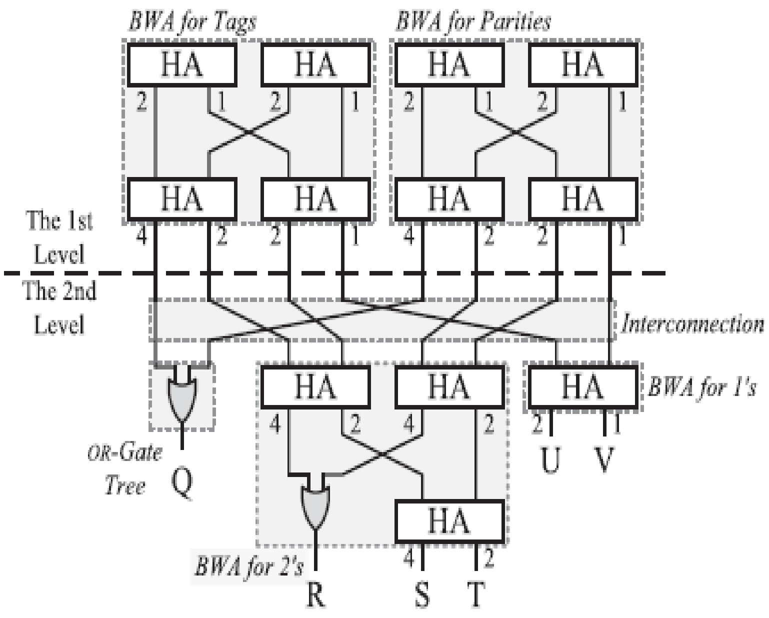


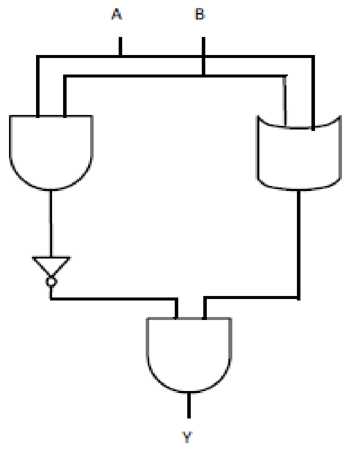
Fig: 5 First and second level circuits for (8,4) code.

Taking the outputs of the preceding circuits (BWA), the decision unit finally determines the incoming tag matches the retrieved codeword by considering the four ranges of the Hamming distance. The decision unit is infact a combinational logic of which functionality is specified by a truth table that takes the outputs of the preceding circuits as inputs. For the (8, 4) code that the corresponding first and second level circuits are given above, the truth table for the decision unit is described in Table 1.

Table I: Truth Table Of The Decision Unit

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

The proposed BWA is developed with the help of modified XOR gate (XORM) and modified half adder (HAM). XORM has 1 gate less than the conventional XOR gate of 5 gates (AND-OR-NOT implementation) as in Fig.5. HAM has 2 gates less than the conventional half adder as shown in Fig.6.As the number of gates reduce in the basic building blocks of the proposed BWA area is also reduced.The block diagram representation of modified BWA is sown in Fig:8.



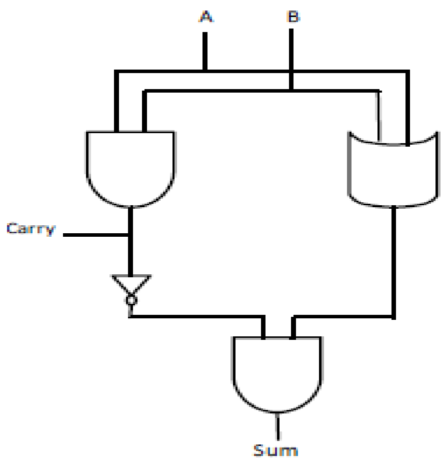


Fig:6 Modified Xor Gate Fig:7 Modified Half Adder

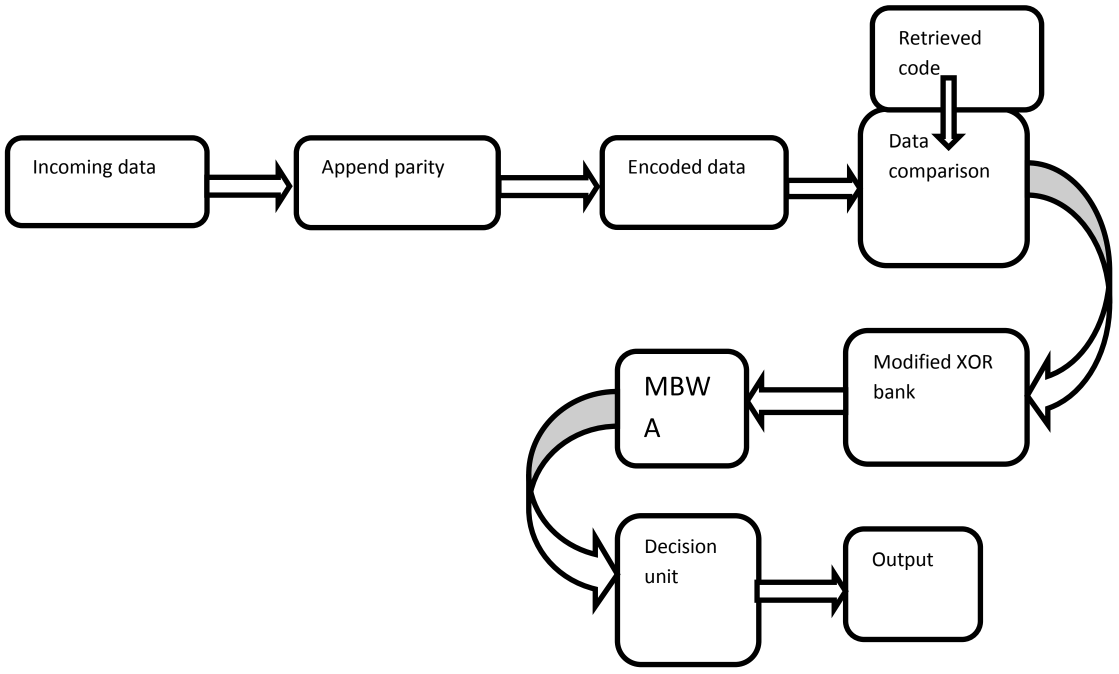


Fig:8 Block Diagram Representation Of Modified BWA

IV. EVALUATION

The proposed algorithm is coded in Verilog and simulated using ModelSim and Xilinx ISE 6.1isimulator and implemented in Spartan 2E. The results obtained are compared with the existing version of the technique.

Table II: Performance Analysis Table

Architecture	No. of slice used	Delay (ns)	Power consumptio-n(mW)
Direct compare	785	43.766	1741
BWA	612	43.17	1259
Modified BWA	591	36.233	982

From the table we can see that modified bwa has lower power consumption,delay and no.of slice used. The new approach saves roughly 40% and 35% in total gate counts. With lower gate count and less area, the routing and interconnect complexity should be lower resulting in smaller routing area and shorter routing latency.

V. CONCLUSION AND FUTURE WORK

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 stored data if a certain number of erroneous bits are corrected. 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. In the proposed architecture we deduct and correct single error and deduct double error in future Error correction of double adjacent errors will be included and compared its performance.

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 Encoded With 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 "Circuitand physical design implementation of the microprocessor chip for the Enterprise system," *IEEEJ. 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 third generation 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," *inIEEE ISSCC. Dig. Tech. Papers*, Feb. 2003, pp. 246– 247.
[8]. C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 124–134, 1984.
[9]. G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault tolerant solid state mass memory for space applications," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 4, pp. 1353–1372, Oct. 2005.
[10]. M. Y. Hsiao, D. C. Bossen, and R. T. Chien, "Orthogonal latin square codes," *IBM J. Res. Develop.*, vol. 14, no. 4, pp. 390–394, Jul. 1970.
[11]. X.Zhou,k.y.Lim and D.Lim, "A Simple and Unambiguous Definition of Threshold Voltage and its Implication in Deep Submicron MOS device modeling ,IEEE transactions on electron devices vol 46,no.4,April 1999.
[12]. James T. Kao and Anantha P. Chandrakasan, " Dual-Threshold Voltage Techniques for Low-Power Digital Circuits", IEEE Journal of solid state circuits,vol.35,no.7,july 2000.

.