ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

A Promising Solution for Comparison of Data Coded With Error-Correcting Codes

Shanitha v.k[1], Jobins George[2]

PG scholar, Dept. of ECE, Ilahia Engineering College, Muvatupuzha, Kerala, India[1]

Assistant Professor, Dept. of ECE, Ilahia Engineering College, Muvatupuzha, Kerala, India [2]

ABSTRACT*:* Data comparison circuit usually resides in the critical path of components that are intended to increase the system performance. So these circuits should be designed to have very low latency and complexity as possible to increase the overall performance of the system. A promising solution for comparison of data which is protected by Error-Correcting Codes (ECC) is presented in this brief to reduce latency and complexity. A practical ECC codeword is usually represented in systematic form in which data part and parity part are completely separated. This property of systematic codeword is used for proposing a new architecture for data comparison. In addition, a new butterfly-formed weight accumulator (BWA) is introduced for the efficient computation of the Hamming distance to further reduce the latency and complexity. This BWA uses a modified half adder (HA) circuit and an XOR design to further reduce area and latency. The implementation result also shows drastic reduction in area and latency using this proposed method.

KEYWORDS: Error Correcting Codes, Butterfly-Formed Weight Accumulator, Half Adder

I.INTRODUCTION

Data comparison circuit is an integral part of computing systems. It is mainly used for tag matching in a cache memory and in translation look-aside buffer (TLB) unit to speed up virtual to physical address translation. For convenience only tag matching in a cache memory is addressed in this paper. Currently microprocessor caches are set-associative caches. Cache is a component that stores data that might be results of an earlier computation or duplicates of data stored elsewhere for faster access. It consists of a tag directory which stores tag address and data array. These tags are compared with the tag field of incoming address to check for match. A cache 'hit' occurs if incoming or requested data is found in cache else cache 'miss' occurs. Cache hits are served by reading data from cache, which is faster than recomputing a result or reading from a slower data store. The more requests can be served from the cache faster the system performance. Thus it is important to implement data comparison circuit with low complexity and latency to improve the overall system performance.

Caches in modern microprocessors are protected by ECC. ECC is used in most of computers where data corruption cannot be tolerated. If a memory structure is protected by ECC, data is encoded first and the encoded tag which contains tag information and check bits are written into the memory array. This encoded data should pass through ECC Decoders and ECC correction circuits before comparing it with incoming tag field. Thus latency is increased due to the added ECC logic. ECC Decoding and correction is also a complicated process, thus the overall complexity of data comparison circuit is further increased.

Direct compare method is a sophisticated solution for data comparison problem. Here the incoming tag is encoded first and compared with the n-bit retrieved codeword. Thus the complex decoding procedure is eliminated in this method. The major disadvantage of this method is it does not check whether both retrieved data and incoming data is exactly the same i.e., only relative value of incoming data is important for deriving comparison. Absolute value of stored information is not that important.

Absolute comparison of data demands an additional efficient and fast performing circuit to compute the Hamming Distance between the two codewords i.e. the bit difference between two codewords. Based on the complexity of the Hamming Distance computing circuit different data comparison architectures were introduced. In this paper we renovate the existing circuits by resolving its drawbacks for a fast and efficient operational data comparison circuit. Thus a low-complexity low- latency hardware architecture for data comparison is proposed here.

DOI: 10.15680/IJIRCCE.2015. 0310014

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

II. LITERATURE REVIEW

Despite the need for sophisticated designs for reducing the latency and complexity of data comparison circuits, the works that cope with these problems are not widely known in literature since it has been usually treated within industries for their products. Conventional decode-and-compare architecture and encode-and-compare architecture based on direct compare method were the main existing methods used and the basic circuits employed are discussed here in detail. For the sake of concreteness, only tag matching performed in a cache memory is discussed.

1. CONVENTIONAL DECODE-AND-COMPARE ARCHITECTURE

Conventional decode-and-compare architecture is depicted in fig.1. The n-bit retrieved code is read from tag directory and it is decoded to give k-bit of information. The incoming tag is then compared with this k-bit of decoded data using a comparator and match/mismatch results.

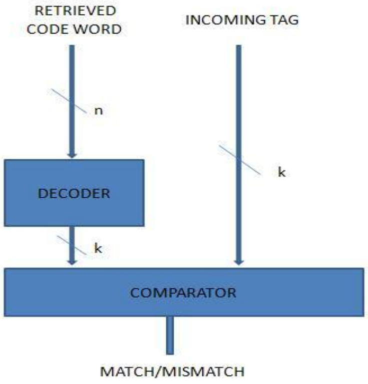


Fig.1. Decode-And-Compare Architecture

Fig.2 shows extraction of k-bit tag from the n-bit codeword. Data comparison is done between one input data and multiple stored data. Thus this method needs multiple ECC Decoding and Correction logic which makes it more complex. Latency includes time for tag directory access, decoding and correction, and time for comparison, thus makes the operations slow. The cache 'miss' causes the incoming tag to be encoded by the 'ECC Gen' logic and is stored in tag directory. MCA signal refers to the Machine Check Architecture in which the microprocessor has detected an uncorrectable error and the system needs to take action.

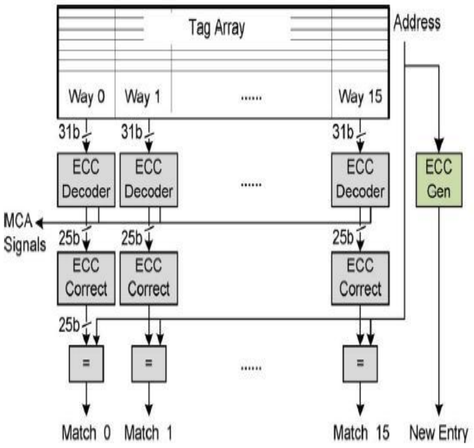


Fig.2. Cache Access Using Decode-And-Compare

DOI: 10.15680/IJIRCCE.2015. 0310014

2. CONVENTIONAL ENCODE-AND-COMPARE ARCHITECTURE

To resolve the drawbacks of encode-and-compare architecture, decode-and-compare architecture were introduced. The n-bit retrieved codeword from tag directory is compared directly with the encoded n-bit incoming tag as in fig.3.

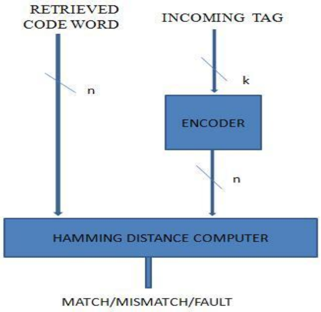


Fig.3. Encode-And-Compare Architecture

As depicted in fig.4 retrieving information from tag directory is performed parallel with the encoding of incoming tag. Since the complex decoding and correction is eliminated this method proves simpler than decode-and-compare architecture. Here the latency includes time tag directory access and tag comparison. As the decoding and correction process is eliminated the overall latency is improved. Only one ECC decoder and correction circuit for' ECC gen' is needed in this case.

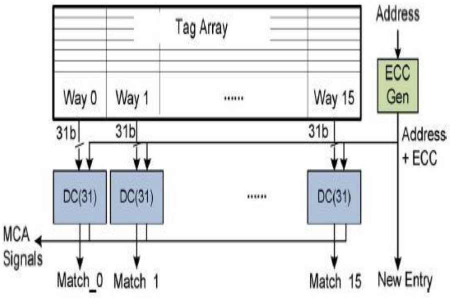


Fig.4.Cache Access Using Encode-And-Compare

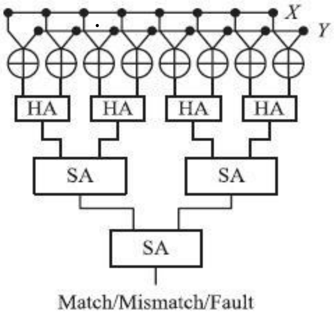


Fig.5. SA-Based Hamming Distance Computing Circuit.

Comparing two tags means to check whether the two codeword are same or to find the number of different bits between them. Hamming distance *d* between the codewords are computed and match/mismatch/fault results according to the value of *d*. This method necessitates the need for Hamming distance computing circuit for finding *d* which is shown in fig.5.The circuit consists of three layers. The first layer is an array of XOR-gates to find the bitwise difference between codewords *X* and *Y* . The second layer consists of HAs or summing circuit to count the number of 1's in the output of first layer. Final layer uses saturate adder (SA) for direct compare design. The output value of this adder has an upper bound and any value exceeding its extreme values will be discarded. This compulsory saturation requires an additional circuitry which makes SA complex.

III. BASIC IDEA OF PROPOSED METHOD

A practical ECC codeword is usually represented in systematic form, Fig.6 in which data part and parity part are completely separated. This property of systematic codeword is used for proposing a new architecture for data comparison with low complexity and latency.

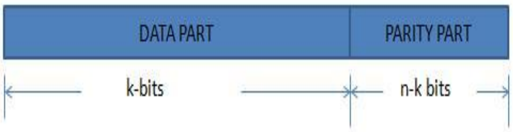


Fig.6.Systematic Representation of An ECC Codeword

A. DATAPATH DESIGNS FOR SYSTEMATIC CODES

In direct compare method the comparison of two codeword is performed after decoding the retrieved code word in decode-and-compare architecture and after encoding the incoming tag in encode-and-compare architecture. It does not consider the systematic representation of ECC codeword in which data and parity bits are completely separated as in Fig.7. In BWA based architecture the data part of retrieved codeword is compared with the incoming tag first and then the n-k parity bits are compared after encoding the incoming tag. As in Fig.8, the k-bit comparison and n-k bit encoding can be performed simultaneously, thus reducing the overall latency.

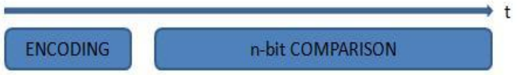


Fig.7. Timing Diagram Of Tag Match In Direct Compare Method.

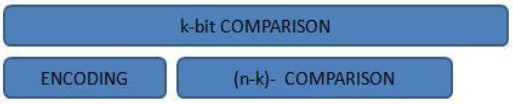


Fig.8. Timing Diagram Of Tag Match In Proposed Architecture.

B. ARCHITECTURE FOR COMPUTING HAMMING DISTANCE

The butterfly-formed weight accumulators (BWAs) based hamming distance computing circuit to improve the latency and complexity is used. The block diagram of this proposed architecture is shown in Fig.9. It contains BWA composed of HAs and or-gate trees. Each XOR bank produces the bit-wise difference between data bits or parity bits. The output of XOR bank is given to corresponding BWA for parities and tags to count the number of 1's i.e to compute hamming distance. Each BWA is composed of numerous stages of HAs where each output bit of HA is linked with a weight. Each HA has the sum or carry of previous stage as input and outputs the sum and carry to next stage as shown in fig.10. At the last stage the number of 1's among the input bits *d* can be calculated as

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

If we know the range of *d*, the circuit can be simplified. For example if the maximally correctable error is 1, two or more than two1's among the input can be regarded as the same. Thus in this case we can substitute a simple OR-gate for several HAs. In the interconnection output from OR-gates and weight bits from HA trees are fed into the related handing out elements at next level. Considering the outputs of the previous levels, the decision unit determines whether there is match between incoming and retrieved codeword depending on the value of *d*. The decision unit is a combinational logic whose utility is specified by a truth table. First and second levels of (8,4) code and its truth table is shown in Fig.11 and Table.1 for better understanding.

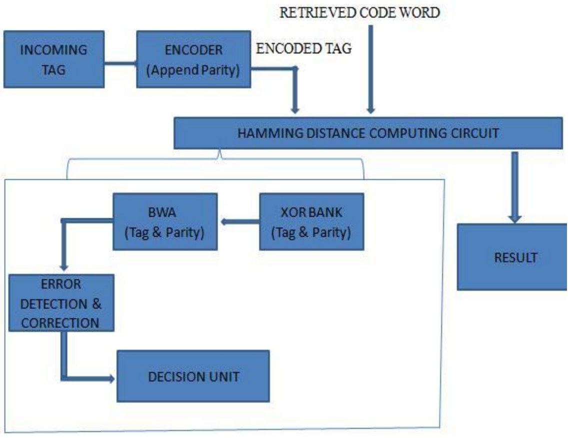
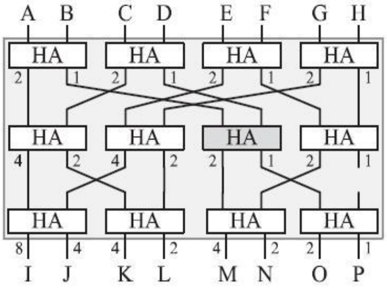
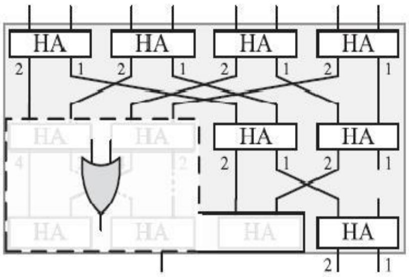


Fig.9. Block Diagram Of Proposed Architecture



(a)



(b)

Fig.10. Proposed BWA (a) Proposed Structure (b) New Revised Structure.

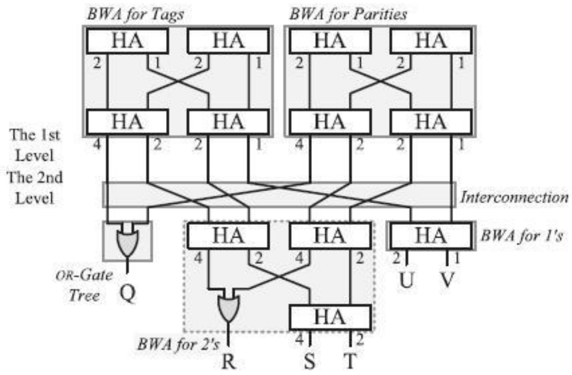


Fig.11. First and second Level Circuits For (8,4)Code

Table.1 Truth Table Of Decision Unit For (8,4) Code

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

For improving performance parameters the proposed BWA structure is developed using modified HAS and XOR gates with less number of gates. The existing BWA structure uses XOR gates designed using conventional AND-OR-NOT implementation. Also conventional HAs using AND gate and XOR is used. Both conventional HA and XOR gate is depicted in Fig.12 and Fig.13.

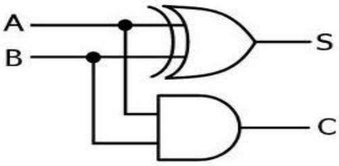


Fig.12. Conventional HA Circuit

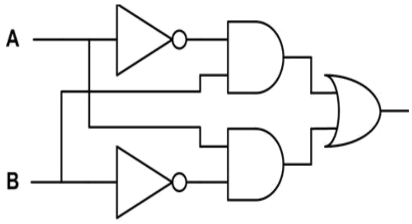


Fig.13. Conventional XOR Gate Design

In this proposed structure modified XOR gate is used which uses 1 gate less than the conventional design. The HA circuit is also modified with 2 gates less than conventional implementation. Both modified HA circuit and XOR gate are shown in fig.14 .and fig.15. As the BWA structure consists of multiple stages of HA circuits the reduction in number of gates in HA reduces the overall area thus complexity thus latency of the entire architecture.

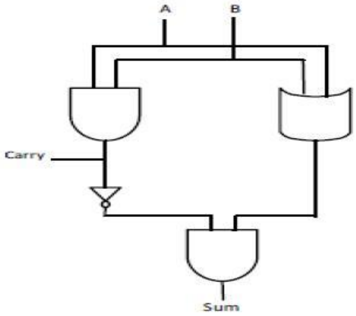


Fig.14. Modified HA Circuit With Less Number Of Gate

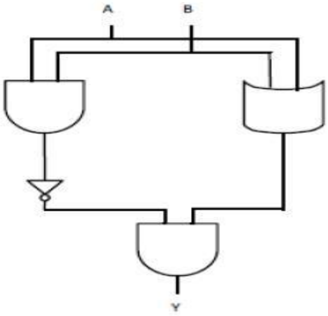


Fig.15. Modified XOR Gate Design With Less Number Of Gate

IV.EVALUATION

Latency and hardware complexity of (24,18) code for different architectures 1) conventional decode-and-compare 2) encode-and-compare SA 3) BWA existing 3) proposed method are given in Tab.2.Complexity is described by counting number of slices used during synthesis and latency is measured in nanoseconds. The table value shows the proposed architecture is effective in improving latency and hardware complexity.

DOI: 10.15680/IJIRCCE.2015. 0310014

Table.2. Evaluation Results of Different Architecture

Architecture	**Number of slices used out of 1728**	**Delay in nanoseconds**	**Power Consumption in Mw**
Decode-and-compare	1365	61.02	1921
Encode and compare SA based	786	49.74	1717
BWA existing	561	39.84	932
BWA modified	486	36.06	815

From the results it is undoubtedly proved that the latency and complexity of the proposed architecture has been improved to a greater extend.

V.CONCLUSION

A new architecture has been proposed to reduce the latency and complexity for matching of data encoded with hard systematic error correcting codes. Replacing the conventional HA circuits and XOR gates with designs consisting of reduced number of gates drastically reduces the area of the proposed architecture. The reduction in area in turn reduces the complexity as well as latency. As the proposed architecture is effective in reducing latency and complexity it can be regarded as a promising solution for comparisons of ECC protected data.

REFERENCES

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

[2]Pedro Reviriego, Salvatore Pontarelli, Juan Antonio Maestro, and Marco Ottavi, “A Method to Construct Low Delay Single Error Correction Codes for Protecting Data Bits Only,” IEEE Trans.Computer-Aided Design of Integrated Circuits and Systems., vol. 32, no. 3, pp. 479 - 483, March 2013. |

[2] R. W. Hamming,“Error detecting and error correcting codes,” Bell Syst. Tech. J., vol. 29, pp. 147–160, Apr. 1950.

[3] M. Tremblay and S. Chaudhry, “A third-generation 65nm 16- core32-thread plus 32-scout-thread CMT SPARC processor,” in ISSCC. Dig.Tech. Papers, Feb. 2008, pp. 82– 83.

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

[5] International Journal of Science and Applied Information Technology (IJSAIT), Vol. 3 , No.4, Pages : 33 - 38 (2014)Special Issue of ICECT 2014 - Held during September 01, 2014 in The Golkonda Hotel, Hyderabad, India 33 Reduced Area and Low Power Square Root Carry Select Adder.