International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 5, May 2016

ASIC Implementation of Data Comparison Circuit for Cache Memory

[1] Basavaraj Mirji [2] R Pramod [3] Dr. V. Anandi
[1][2] M.Tech Student [3] Associate Professor
Department of ECE, M. S. Ramaiah Institute of Technology, Bangalore India
[1] mirji07@gmail.com [2] pramodr91@gmail.com [3] anandi.v@msrit.edu

***Abstract*— In current scenario, computing system of microprocessor involves data comparison circuit for matching input data to the stored data in memory. For protection of data and to improve reliability, the recent microprocessor uses error correcting code. In this paper a new architecture for data matching is presented to reduce the delay and hardware circuit complexity. Taking into account the property of the systematic codeword, the deliberate shape comprising of information and parity bits independently, so parallel examination of information and parity bits diminishes the delay of the overall circuit.**

***Keywords*—Data comparison, set associative cache, error correcting code (ECC), ASIC, Hamming Distance**

I. INTRODUCTION

Information comparison circuit is broadly utilized as a part of cutting edge modern microprocessor to perform operations, for example, cache memory tag coordinating. Other than the output of information comparison circuit decides the stream of succeeding operations in pipeline. Along these lines circuit should be planned to have low hardware complexity and low delay. To improve the dependability and integrate of memory structures the vast majority of cutting edge microprocessor utilizes error correcting code. On the off chance that cache memory is secured with ECC, an approaching tag is encoded first and the whole codeword including parity bits are stacked into memory cluster. At the point when codeword is stacked from memory, to acquire the information the codeword has be decoded and remedied if blunders are distinguished.

The latest approach for coordinating issue is the direct compare design [2], which encodes the approaching tag and afterwards it compares with recovered tag from cache memory. Therefore this method eliminates decoding from complex path. While performing comparison this technique does not inspect whether the approaching information is precisely same as the recovered information. Rather it figures out the hamming distance between the recovered information and approaching information which is encoded also. In view of the output of the hamming distance circuit the decision unit decides the error correction and detection range. The saturate adder (SA) was introduced in [2] for figuring the hamming distance. The output of saturate adder is not greater than the number of perceivable errors by more than one.

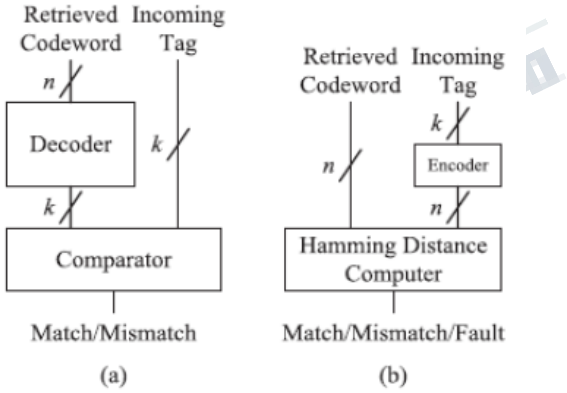

Fig. 1, (a) Conventional Decode and compare architecture
(b) Direct compare architecture

II. PREVIOUS WORK

This section describes two data matching techniques such are conventional decode compare architecture and direct compare architecture [2].

A. Conventional Decode and compare architecture

In decode and compare method, the recovered codeword is decoded first to separate the data and parity bits. The data field is then compared with incoming tag to figure out the labels are matched or not. In this method the recovered tag ought to experience decoder before comparing with the incoming tag, the critical path is too long in a handy cache memory intended for rapid get to.

The decoder is more complicated processing element and the complexity of decoder is not negligible.

B. Direct compare architecture

The latest method for information comparison is direct compare technique [2]. This technique encodes the approaching tag and afterwards compares with the recovered tag that has been encoded too. Hamming distance d between the two codewords looks at the what number of bits of two codeword's vary. **Since this** strategy needs to compute the hamming distance, [2] introduced a circuit committed for calculation.

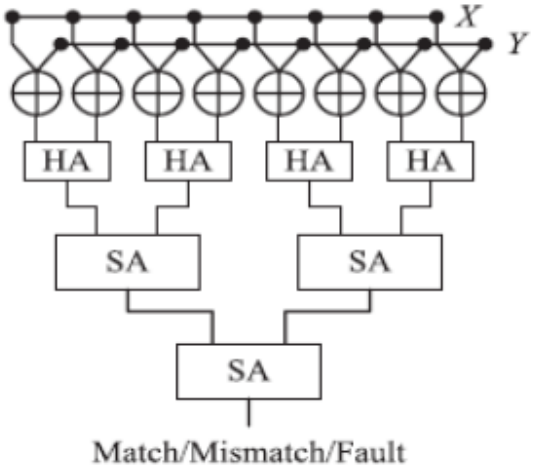

Fig. 2. SA Based hamming distancs computation block

The SA based hamming distance calculation circuit first performs XOR operation for each pair of bits in codeword. The XOR operation creates bitwise distinction vector of two codeword's. Then the responses in two contiguous piece vectors are processed by half adders at the second step.

The quantity of 1's in bit vector are ascertained by going through the accompanying SA tree. The output estimation of last saturate adder demonstrates the scope of d.

Let r_{max} and t_{max} indicate the quantities of maximally detectable errors and correctable errors, respectively. The cases are condensed as

1. If d=0, input tag matches retrieved tag
2. If $0<d<t_{max}$, input tag will match retrieved tag provided at most t_{max} errors in retrieved tag are corrected.
3. If $t_{max} <d<r_{max}$, retrieved tag has detectable but uncorrectable errors.
4. If $r_{max}<d$, input tag does not match retrieved tag.

On the off chance that the approaching tag has no errors, we can see the two labels as coordinated then the hamming distance d is in either first or second range.

III. PROPOSED ARCHITECTURE

To lessen the hardware complexity and output delay of information comparison circuit, this section exhibits another design by considering the attributes of systematic code. The direct compare design performs comparison of two code words after the approaching tag is encoded. In this way basic way comprises of arrangement of encoding and the n-bit examination operation. In any case, direct compare technique [2] did not consider the way that, the ECC codeword is of efficient structure in which the information fields and parity fields totally isolated. In the event that the information field of retrieved codeword matches with the approaching label field which is encoded later. At that point the information field of approaching tag and recovered tag is immediately compared while the parity part get to be accessible strictly when the encoding is finished.

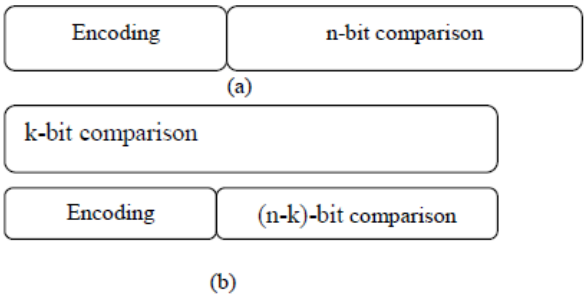

Fig. 3. Tag comparison timing diagram in (a) direct compare technique [2] (b) parallel comparison technique

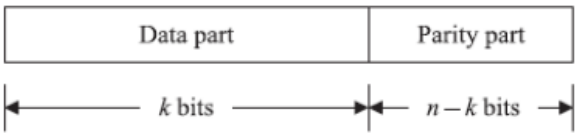

Fig. 4. Systematic ECC codeword representation

Proposed Hamming Distance Computation Circuit

The proposed technique is shown in figure 5. The proposed circuit enhances the complexity of hardware and output delay of the hamming distance calculation block. It comprises of numerous phases of full adder as appeared in figure, where output bit of full adder is related weight. The essential function of modified hamming distance block is to tally the quantity of responses (1's) among its information bits. Note that in figure 5, the sum and carry bits are separately processed by full adder and there is no overlap in between. The both inputs of a full adder in a stage, with the exception of first stage, are either sum bits or carry bits figured in upper stage. The quantity of 1's among bits in the way reaching the full adder is equivalent to the weight of the result bit.

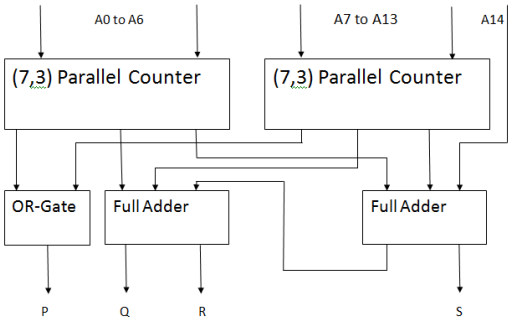

Fig. 5. Hamming Distance Computation Block for (15,11) codeword

The hamming distance calculation circuit comprises of two sections. The first part is exhibit of bitwise comparator, it creates bit vector representing bitwise distinction between the two code words. The second part is a modified parallel counter which tallies the quantity of 1's from the outputs of XOR array from the first step. The output weight bits of parallel counter stage are associated with the second level as indicated by their weights. The second stage of parallel counter generates the output from the several weight bits from full adder tree and OR-gate tree. As weight bits connected with fourth territory are all ORed in altered parallel counter. The output of OR-gate tree is associated with resulting OR-gate tree at second level.

The decision unit decides the incoming codeword matches with retrieved codeword based on the output of the hamming distance value. The decision unit is combinational circuit and the operation of it is indicated by truth table that takes the output of the hamming distance circuit as inputs.

The truth table for decision unit and modified hamming distance computation block for (15,11) systematic codeword is shown in figure.

TABLE 1
DECISION UNIT FOR (15,11) CODE

P	Q	R	S	Decision
0	0	0	X	Match
0	0	1	X	Fault
0	1	0	0	Fault
0	1	0	1	Mismatch
0	1	1	X	Mismatch
1	X	X	X	Mismatch

IV. APPLICATION TO CACHE MEMORY

Most present microprocessors caches are set associative caches. The cache memory has tag directory and a information array. The tag index stores label addresses which are utilized to show which portion of memory is put away in information array. At the point when an access is made to the cache the set location bit of the whole address is utilized to record into the label catalog and set of labels are read. These label fields are compared with label field of incoming address to check whether there is a match.

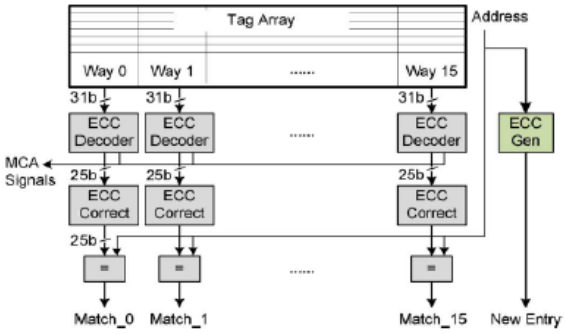

Fig. 6. Data operation in cache tag

In the event that the label index is secured with ECC, decode and compare technique is to read the encoded tag from the label catalog first. As shown in figure 6, recovered information experiences ECC decoders and ECC correction before it is compared with the label field of approaching field. In the event that the approaching tag is does not match to any of the stored labels, a cache miss happens. Cache total access time is the whole of the label index access time, decoder and rectification time and the time required for the label correlation.

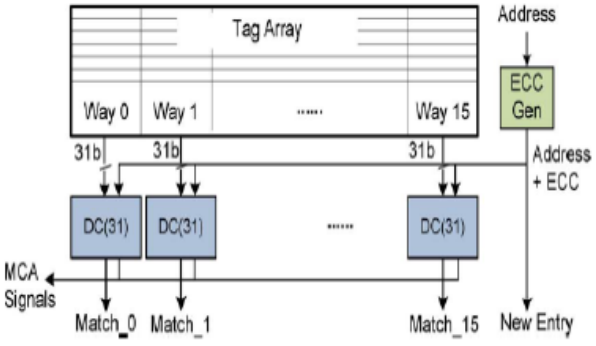

Fig. 7. Data operation with the direct compare

Figure 7 outlines the cache access utilizing the direct compare method. The recovered label directory is contrasted with the approaching label field after it is encoded. The encoding of the approaching tag can be performed in parallel with label index access. The

decoding and correction block is expelled from the basic way.. Along these lines the aggregate access time is the whole of the label index access time and label comparison. Utilizing machine check design the chip recognizes the uncorrectable blunders and the framework needs to make a move.

V. RESULTS

The design of decode and compare architecture, Direct compare architecture and the proposed architectures have been simulated using Cadence NC verilog simulator. The code words chosen for simulation are (15,11) and (31,26) linear systematic codes .

In order to compare the complexity overhead and power consumption, the all architectures are synthesized using Cadence Encoder RTL complier tool. Table 2 shows that the hardware complexity and power consumption of proposed architecture is less than the decode and compare architecture.

ECC Code Words	Architecture	Cell Area	Total Power (nW)
(15,11)	Direct compare	389	24373
	Proposed	354	16718
(31,26)	Direct Compare	960	75554
	Proposed	877	44039

VI. CONCLUSION

To decrease output delay and hardware complexity of comparison circuit another design has been exhibited for coordinating information ensured with an ECC. The new architecture looks at whether the approaching codeword matches with stored codeword with single error correction and two error detection. The examination of the code words is parallelized with the encoding procedure to diminish the delay of circuit. The parallel operations are performed taking into account the way that deliberate codeword has separate information fields and equality fields. The proposed hamming distance computation block reduces the hardware complexity.

REFRENCES

[1] 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] Wei Wu, D Somasekhar, Shih.-Lien Lu "Direct compare of information coded with error correcting codes", IEEE Trans. Very Large Scale Integr. Syst., vol. 20, no. 11, pp. 2147–2151, Nov. 2012.

[3] Current K W, Mow, Douglas A, "Implementing Parallel Counters with Four Valued Threshold Logic", IEEE Transactions on computer, August 2006

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

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

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

[7] Caxton C, Foste, Fred D. Stockan, "Counting Responders in an Associative Memory", IEEE Transactions on computer, December 1971.

[8] Brown, Frank M, "Weighted Realizations of Switching Functions", IEEE transactions on computer, December 1975.