

## Data Comparison With Error Correcting Codes

<sup>1</sup>P.V.L Prathusha and <sup>2</sup>R.Muthaiah

<sup>1,2</sup>*School of computing, SASTRA University*

<sup>1</sup>*E-mail: pvlprathusha@gmail.com*

### Abstract

This paper proposes a new method to compare data stored in cache memory with the incoming data. It is mainly applicable in computer memories, communication systems, etc. This can be achieved with the help of Error Correcting Codes (ECC) so that data can be sent from transmitter to receiver in an error free manner. This method parallelizes both the data and the parity bits separately. BWA (Butterfly Weighted Accumulator) are used in which multiple half adders are connected together to calculate sum and carry separately. This improves the computation of hamming distance. This further reduces latency and complexity thereby improving system performance.

**Keywords:** BWA, ECC, latency.

### 1. Introduction

The information that is to be sent from transmitter to receiver is to be first sent from source to the encoder. Here the information is encoded in which analog to digital converters are used and is converted into codeword by adding redundancy bits [6]. The encoder divides the information into message blocks of length 'm' of codeword length 'n'. Then the information is sent through the channel and is finally decoded by the decoder at the receiver side. ECC are used to protect data during transfer that improves reliability and is applicable in CD players, high speed modems, flash memory etc. Graded bit ECC are used in flash memory which increases the lifetime of the device by eliminating the errors [4].

Data comparison is necessary especially in tag matching in cache memory and in virtual to physical address translation in translation look aside buffer [5]. The TLB's translate from virtual address to physical address.

Mainly this paper focuses on the application related to tag matching in cache memory. When a processor wants to perform some operations like read or write to memory, it initially checks whether the data to be written is present in cache or not. If so, it immediately takes from cache, which is much faster than reading from or

writing to the main memory. Caches have limited capacities so, they are shared by data and instructions from the main memory. Cache sharing is allowed by storing upper bits of memory address and to index caches, lower bits of address are used thereby improving the performance. The cache contains tag, index and offset. The tag allows the cache to translate from cache address to a unique CPU address. A cache block is a basic unit for cache storage.

The distance between two codeword's is termed as hamming distance. The number of non-zero one's is termed as hamming weight. Hamming weight comparators are less complex and faster and find applications in pattern matching/recognition, data compression, digital filtering, digital neural networks, etc [7]. Hamming codes are double detecting codes and single error correcting codes. The mostly used codes for data protection in a memory of a microprocessor are BCH codes and hamming codes which fall into category of linear block codes[3]. Hamming Codes are mainly applicable in digital communication and data storage systems (eg: in RAM). For 'p' as a positive integer, there can be some variables to be defined like:

Code length:  $n = 2^p - 1$ ,

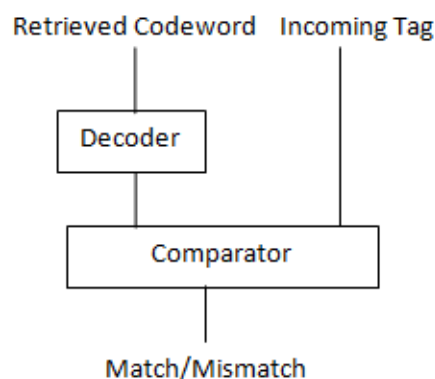
Number of information or message bits:  $m = 2^p - p - 1$ ,

Number of parity bits:  $n - m = p$ ,

Minimum distance: 3

## 2. Related Works

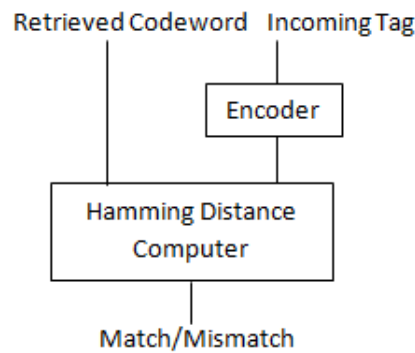
- a) Decode and Compare architecture: It consists of a decoder and a comparator as shown in figure1. Initially the retrieved codeword is decoded first and is compared with the incoming tag. Match or a mismatch is found. The circuitry of this is costly and is more complex. This drawback can be overcome with the use of encode and compare architecture [1].



**Fig 1: Decode and compare architecture**

- b) Encode and Compare architecture: To overcome the drawbacks of decode and compare architecture, encode and compare architecture is used. It is a Direct Compare method that uses an encoder and a hamming distance computer as

shown in figure 2 [2]. Initially the incoming tag is encoded to a codeword 'X' and is sent to the hamming distance computer. Here both the retrieved or stored codeword 'Y' and the codeword 'X' are compared. Saturate Adders (SA) are used to compute the hamming distance. This is done by performing XOR operation for every pair of bits in which half adders are used to count the number of 1's. There are two conditions to be observed: the input codeword is always correct and the retrieved codeword matches the input codeword if and only if their hamming distance is equal to or less than  $t_{\max}$ , where  $t_{\max}$  denotes the number of correctable errors. Finally a match or a mismatch is found. In this method the ECC decoding is removed from the critical path.



**Fig 2: Encode and compare architecture**

For (4,2) saturate adder, sum and carry can be calculated as:

$$C_x = C_a + C_b + S_a S_b$$

$$S_x = S_a \text{ XOR } S_b + C_a S_a + C_b S_b + C_c S_c$$

For suppose, two codeword's X and Y are to be compared. For this, hamming distance 'd' is computed. Let  $r_{\max}$  and  $t_{\max}$  denote detectable errors and correctable errors respectively and following 4 cases are classified according to the range of d.

- 1, X matches Y exactly if  $d=0$ .
- 2, X matches Y provided at most t errors in Y are corrected if  $0 < d \leq t_{\max}$ .
- 3, Y has detectable but uncorrectable errors if  $t_{\max} < d \leq r_{\max}$ .
- 4, X does not match Y if  $r_{\max} < d$ .

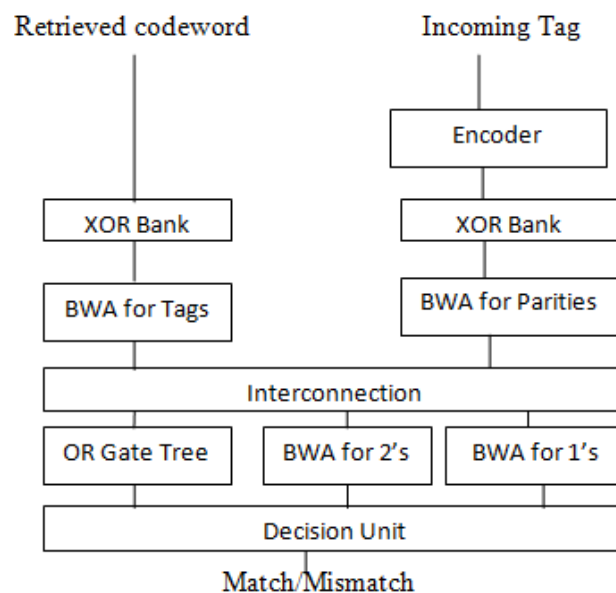
### 3. Proposed Work

The circuitry to be implemented usually resides in the critical path. This affects the system performance. Therefore a new method is proposed to improve the system performance in which BWA's are used to compute the hamming distance faster. Further to decrease the latency, parallel processing is implemented.

- a) Random Code Generator: This implements the use of LFSR (Linear Feedback Shift Register) in which codeword of length 8 and the incoming tag of length 4

are generated. This uses D flip flops and XOR gates. LFSR counter's are of two types: Up counters, down counters. Up counters use XNOR gates in feedback whereas Down counters use XOR gates in feedback. LFSR is a special type of counter with  $2^n - 1$  state. It is easy even to test for faults.

- b) **Systematic Code Generator:** The codeword is said to be semantic, if the codeword consists of information bits with additional bits called parity bits. A codeword is said to be odd parity if it has odd number of ones in the codeword else it is said to be even parity if it has even number of ones in the codeword [8]. A linear block code is always considered to be equivalent to systematic code. Both the data and the parity bits are computed in parallel, thereby reducing the overall latency and complexity. It contains XOR banks, an encoder, BWA's, OR gate tree and a decision unit as shown in figure 3. Initially the incoming codeword is encoded i.e from 8 bit codeword to 4 bit code and then it is fed to the Butterfly weighted accumulators where tag and parity bits are computed separately and the number of one's are calculated. Faster computation of hamming distance can be achieved with BWA's. This arrangement consists of many half adders (HA) which are connected in a butterfly structure, to calculate sum and carry separately. Range of the hamming distance 'd' can be calculated.



**Fig 3: Systematic Code Generator**

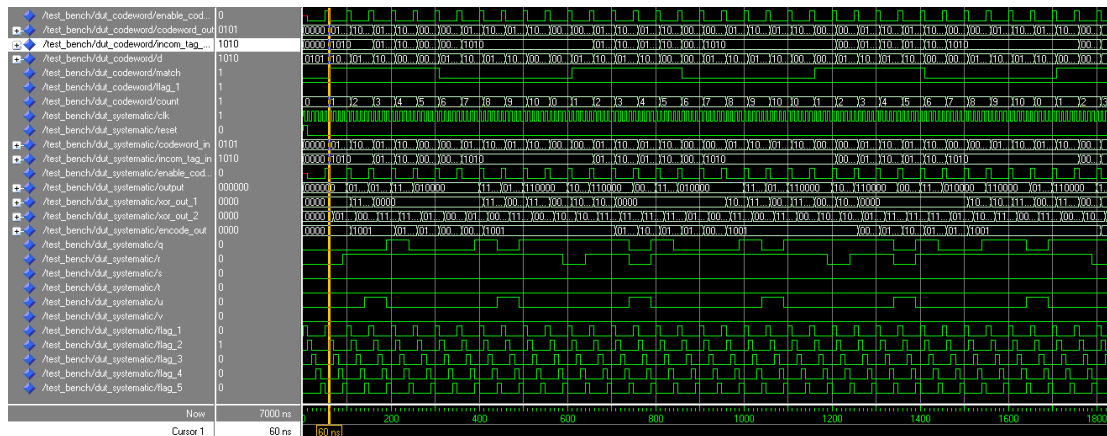
Several half adders can be replaced with an OR Gate tree. The decision unit performs functionality and is specified in the table 1.

**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

**4. Results**

Simulation for systematic code generator for (8,4) code is done in modelsim and the output is observed in below figure 4.



**Fig 4: Output of systematic code generator implementing Parallel Processing**

**Synthesis Report:**

Synthesis is carried in Xilinx tool of Spartan 3e family. The table 2 illustrates various factors that are observed in this tool.

**Table.2: Synthesis Report in Xilinx**

S.No	Parametres	Values
1	Number of slices used	31
2	Power	0.081 W
3	Frequency	264.211MHz
4	Time Period	3.785ns

## 5. Conclusion

A new architecture for systematic code generator is presented here that reduces latency and complexity. System performance is improved. This is mainly applicable in tag matching in cache memory in semiconductor memories. Parallel Processing is enabled for data and parity bits separately for systematic code generator.

## References

1. Byeong Yong Kong, Jihyuck Jo, Hyewon Jeong, Mina Hwang, Soyoung Cha, Bongjin Kim, and In-Cheol Park, July 2014, "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.
2. W. Wu, D. Somasekhar, and S.-L. Lu, Nov. 2012., "Direct compare of information coded with error-correcting codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol.20, no. 11, pp. 2147–2151.
3. S. Lin and D. J. Costello, 2004., "Error Control Coding": *Fundamentals and applications*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall
4. Kedar Karmarkar, DECEMBER 2014, Error Correction Encoding for Tightly Coupled On-Chip Buses", *Member, IEEE*, and Spyros Tragoudas, *Senior Member, IEEE* ,IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 22, NO. 12.
5. Jongmin Lee, Seokin Hong and Soontae Kim, 2011, "TLB Index-based Tagging for Cache Energy Reduction", Department of Computer Science Korea Advanced Institute Science and Technology (KAIST) Daejeon, Korea, IEEE .
6. Jatinder Singh, 2012, "A Comparative study of Error Detection And Correction Coding Techniques", University Institute of Engineering and Technology, Panjab University, Chandigarh, INDIA Second International Conference on Advanced Computing & Communication Technologies.
7. Behrooz Parhami, FEBRUARY 2009, "Efficient Hamming Weight Comparators for Binary Vectors Based on Accumulative and Up/Down Parallel Counters", *Fellow, IEEE* ,IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 56, NO. 2
8. Peter Sweeney, 2002, "ERROR CONTROL CODING ", *University of Surrey, Guildford, UK*, John Wiley & Sons, Ltd ,England.