Low Area and Reduced Delay of Encoded Data Using Modified Bwar

[1]Dr. P. Kishore, [2]K. Ratna Phanitha
[1]Professor, Dept of ECE, VNR Vignana Jyothi Institute of Engineering and Technology, HYD, T.S, INDIA
kishore_p@vnrvjiet.in
[2]PG Scholar, M.Tech, Dept of ECE, VNR Vignana Jyothi Institute of Engineering and Technology, HYD, T.S
phanithakakarala@gmail.com

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. Comparison of data is a technique which fetches data together from various sources and compares it. When this matching differs the stored data, the use of proprietary matching algorithms is used to compare and correct the mismatch result. As per the survey, different researches have been done and still going on till date. 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 code word 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. Furthermore, in a renovated butterfly weighted accumulator (BWAR), a type of reversible logic gate called PERES gate is proposed to modify the half adder with disparate algorithm to compute hamming distance with reduced complexity and improved efficiency with testable feature. The proposed architecture examines whether the incoming data matches the stored data if a certain number of erroneous bits are correct.

I. INTRODUCTION

Data comparison circuit is a logic that has many applications in a computing system. Today in communication system data transferring confronts several problems especially in data loss. Data matching is broadly used in calculating structures to achieve any processes particularly in data matching in a cache memory which turns towards data loss problem. 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. 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 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 code word 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 is because the data should be sent from sender to the receiver's end in the form of text, image, files etc. so, to send it properly the algorithms are designed to protect the lost data. The second cause of data loss is increase in hamming distance due to which complexity and latency agonize. These extra problems can affect information loss, costs time, reliability, power, speed, efficiency, delay etc. Low power, area, delay, testable features has become important in today's VLSI design.

As fresh computers service system error correcting codes (ECC) to defend records then recover trustworthiness, intricate decoding process, which necessity lead the data judgment, lengthens the critical path and worsens the complication above. So, it develops much firmer to see the overhead plan restrictions. The direct-compare-method with the BWA (Butterfly weight Accumulator) architecture has emerged as a promising solution to the data damage problem. Direct compare-method replaces the decoding of data by encoding, as decoding is extra complex than encoding. Also, the HA's from the BWA architecture is replaced from unique reversible logic gate i.e. Peres gate to compute the hamming distance with reduced complexity and latency by BWA architecture. We have offered different types of reversible logic gates also which is used then relevant with this research work i.e. Peres Gate. Reversibility in computing denotes that not any evidence about the computational states can ever be missing and thus the incoming tag can be recovered by means of figuring rearward or un-computing result. This is named by way of a logical reversibility. It further results in less area, improved efficiency, testable feature and low power.

II. LITERATURE SURVEY

Low-Complexity Low-Latency Architecture for Matching of Data Encoded with Hard Systematic Error-Correcting Codes

A new architecture for matching the data protected with an error-correcting code (ECC) is presented in this brief to reduce latency and complexity. Based on the fact that the codeword of an ECC is usually represented in a systematic form consisting of the raw data and the parity information generated by encoding, 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 new butterfly-formed weight accumulator (BWA) is proposed for the efficient computation of the Hamming distance. Grounded on the BWA, the proposed architecture examines whether the incoming data matches the stored data if a certain number of erroneous bits are corrected. For a (40, 33) code, the proposed architecture reduces the latency and the hardware complexity by ∼32% and 9%, respectively, compared with the most recent implementation.

The 65-nm 16-MB Shared On-Die L3 Cache for the Dual-Core Intel Xeon Processor 7100 Series

The 16-way set associative, single-ported 16-MB cache for the Dual-Core Intel Xeon Processor 7100 Series uses a 0.624 m2 cell in a 65-nm 8-metal technology. Low power techniques are implemented in the L3 cache to minimize both leakage and dynamic power. Sleep transistors are used in the SRAM array and peripherals, reducing the cache leakage by more than 2X. Only 0.8% of the cache is powered up for a cache access. Dynamic cache line disable (Intel Cache Safe Technology) with a history buffer protects the cache from latent defects and infant mortality failures.

Circuit and Physical Design Implementation of the Microprocessor Chip for the zEnterprise System

This paper describes the circuit and physical design features of the z196 processor chip, implemented in a 45 nm SOI technology. The chip contains 4 super-scalar, out-of-order processor cores, running at 5.2 GHz, on a die with an area of 512 mm containing an estimated 1.4 billion transistors. The core and chip design methodology and specific design features are presented, focusing on techniques used to enable high-frequency operation. In addition, chip power, IR drop, and supply noise are discussed, being key design focus areas. The chip's ground-breaking RAS features are also described, engineered for maximum reliability and system stability.

Implementation of Systematic Error Correcting Codes for Matching of Data Encoded

A Computing system is one, where an input data compared with a stored data to locate the matching entry. For example, translation look aside buffer and Cache tag array lookup matching. In this paper, we propose new architecture in order to reduce complexity and latency for matching the data protected with an error-correcting code (ECC). It is based on the codeword of an ECC generated by encoding is usually represented in a systematic form and it consists the raw data and the parity information. The proposed architecture parallelizes the comparison of the data and that of the parity information. To reduce the latency and complexity, we propose a new butterfly-formed weight accumulator (BWA) for the efficient computation of the Hamming distance. The proposed architecture checks whether the incoming data matches with the stored data.

A Review of Reversible Gates and its Application in Logic Design

Reversible logic has become one of the most promising research areas in the past few decades and has found its applications in several technologies; such as low power CMOS, nanotechnology and optical computing. The main purposes of designing reversible logic are to decrease quantum cost, depth of the circuits and the number of garbage outputs. The purpose of this paper is to give a frame of reference, understanding and overview of reversible gates**.** In this paper various logic gates and its applicability on logic design have been discussed. Also, a brief framework of comparisons between various reversible circuits is presented on the basis of various parameters.

A FPGA Implementation of Hard Systematic Error Correcting codes-based Matching of Data Encoded Architecture with Low Complexity, Low-Latency

Nowadays to get error free data is toughest task. To verify whether the data is error free or not and decrease overall area and latency the new architecture has been designed which matches the data saved in the system and incoming data using Error Correcting Code (ECC).ECC basically consist of two parts parity part and raw data which are generated by encoder .In this method the matching of the data is made parallel so that we can further decrease the area and one more method has been designed called Buttery weight accumulator which correctly calculates the hamming distance. To still reduce the area a new technique has introduced in which we have used the modified XOR gate.

III. EXISTING SYSTEM

ERROR DETECTION AND CORRECTION

EDAC methods are used to find that the data is error free or is not corrupted, either by noisy channel, by hardware failure or during read-write operation in the memory segment. Various error detection methods exist in the communication system. One method currently utilized to produce reliable memory is the use of Error Correction Codes (ECC) to encode data before it is stored in the memory. Error correction codes take a set of information bits at the producer of the information and create a set of redundant bits based on the information bits. These redundant bits are sent or stored with the original set of information bits. The consumer of the information then uses the redundant bits to determine if any errors have occurred in transmission or storage. In the case of memory, the redundant bits are calculated and stored along with the original bits and then when they are read from the memory they are examined to determine if any errors have occurred between the times the information was stored and the time it was retrieved.

The most common error detecting and correcting scheme being employed are parity bit, CRC, HVD and hamming codes. All these methods are implemented on the second layer of OSI model at Data link layer. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Therefore, the upper layers require error-free transmission between two systems. Almost every application did not work if it receiver data with errors. Applications like voice and video may not get that much affected and may still function well with some error. Data-link layer uses some error control mechanism to ensure that data bit streams are transmitted with certain level of accuracy. But to recognize how errors can be controlled, it is important to know what types of errors may occur.

TYPES OF ERRORS

There may be three types of errors:

SINGLE BIT ERROR

In this a frame consists of only one bit corrupted anywhere throughout.

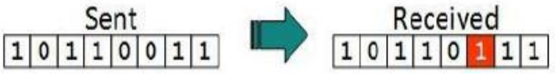


Fig 1: Single Bit Error.

MULTIPLE BIT ERRORS

In this a frame is received with more than one bit in corrupted state.

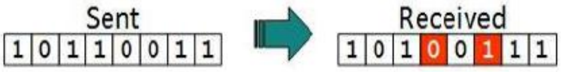


Fig 2: Multiple Bit Errors

BURST BIT ERRORS

In this a frame contains more than one consecutive bits corrupted or more than one bit flips.

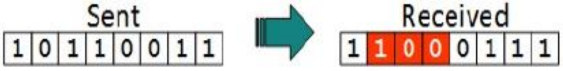


Fig 3: Burst Bit Errors

PARITY

Parity adds a single bit that indicates whether the number of ones (bit-positions with values of one) in the preceding data was even or odd. If an odd number of bits is changed in transmission, the message will change parity and the error can be detected at this point; however, the bit that changed may have been the parity bit itself. The most common convention is that a parity value of one indicates that there is an odd number of ones in the data, and a parity value of zero indicates that there is an even number of ones. If the number of bits changed is even, the check bit will be valid and the error will not be detected.

Moreover, parity does not indicate which bit contained the error, even when it can detect it. The data must be discarded entirely and re-transmitted from scratch. On a noisy transmission medium, a successful transmission could take a long time or may never occur. However, while the quality of parity checking is poor, since it uses only a single bit, this method results in the least overhead.

IV. PROPOSED SYSTEM

DIRECT COMPARE METHOD

Direct compare method 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, this 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 reduces the latency and hardware complexity by resolving the aforementioned drawbacks.

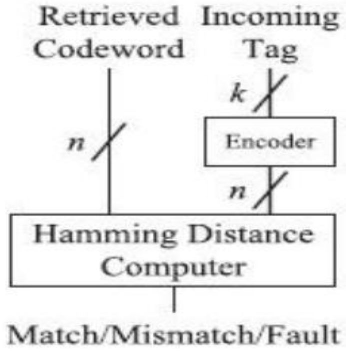


Fig 4: Direct Compare Method

PROPOSED DATA PATH DESIGN:

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 code words. 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.

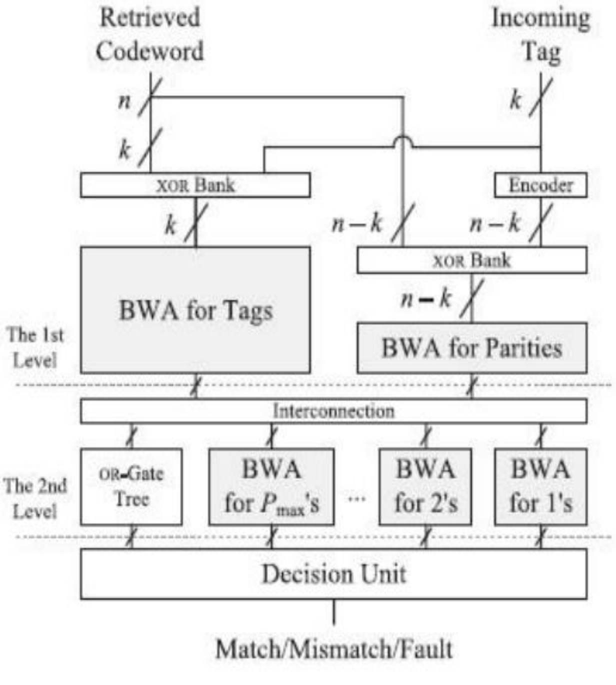


Fig 5: Data Path Design

MODIFIED HALF ADDER DESIGN:

ISSN: 2005-4238 IJAST

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). HAM has 2 gates less than the conventional half adder as shown in Fig. As the number of gates reduced in the basic building blocks of the proposed BWA area is also reduced.

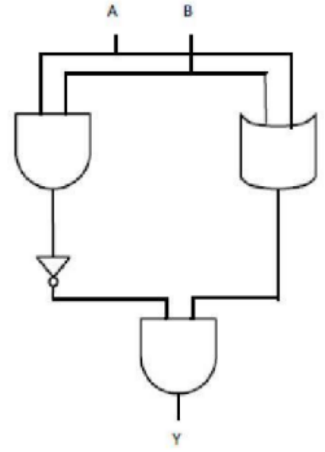


Fig 6: Modified XOR Gate

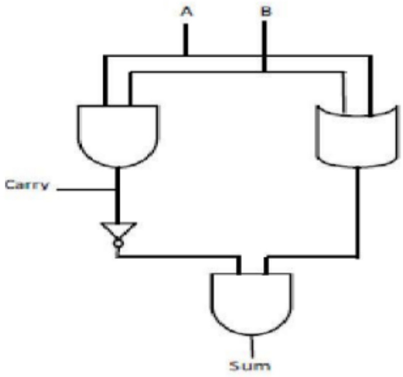


Fig 7: Modified Half Adder

FUNCTIONAL UNITS OF SYSTEMATIC CODES:

Earlier, decode and-compare method was adopted for data comparison. This was a lengthy process of fetching the data, decoding and then comparing with the incoming code word. Additionally, the error was never recovered. So, nowadays, the data is encoded using ECC in computing systems for better reliability, which is of an order separating data bits and parity bits shown in Fig where; n= total number of bits (data part + parity part) k= data part (n-k) = parity part

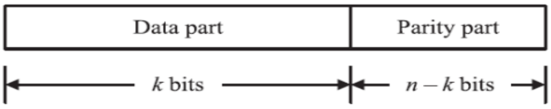


Fig 8: Systematic Representation of An ECC Code Word

Here, data bits of a systematic code word is replica of the arriving code word, which immediately compares while parity becomes available only after the encoding is completed. In this way, k-bits incoming code word is compared earlier the (n-k) bits evaluation i.e. parity parts, ultimately generating parity bits from the incoming code word by encoding which is carry out in parallel with the data comparison.

BWAR FOR HAMMING DISTANCE ESTIMATION:

The proposed architecture is described in Fig. which is grounded on the functional unit design. A multiple BWARs are proposed for estimation of the hamming distance. As in proposed BWAR no additional logic is implemented, this result in low complex circuit. Here the interconnection between the half adders is modified in such a way that it disposes SAs. The BWAR consists of multiple stages of half adders to sum up the difference in parity and incoming code words. Each output of a half adder is associated with a weight. The general BWAR structure taken into reference for this proposed work is mentioned above in Fig. This BWAR efficiently matches the encoded data with ECCs which can be fetched in the memory. In the next stage, the XOR bank is introduced to find the bit changes and BWAR structure is utilized to comp1ute the number of 1s to reduce the hamming distance. BWAR for incoming code words and parity are separately designed in the proposed architecture. Lastly, as per the output of the BWAR the final matching is obtained by the help of a decision unit, indicating match, mismatch or fault, designed with the help of a hamming distance concept. The proposed architecture for the renovated Butterfly Weight Accumulator is drawn in the next Fig

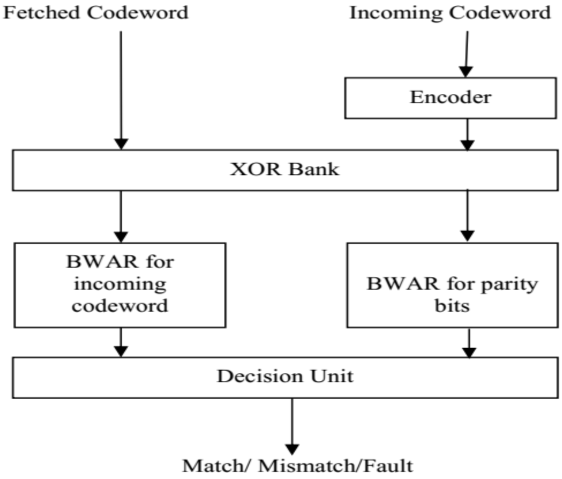
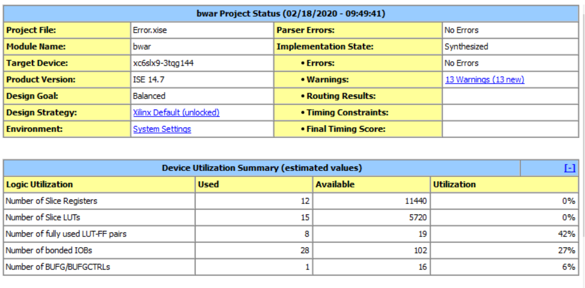


Fig 9: BWAR Based Architecture

RESULTS AND DISCUSSION

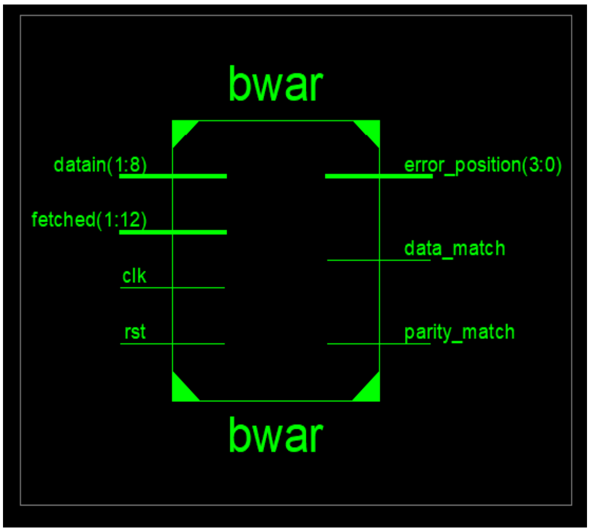
HARDWARE UTILIZATION SUMMARY:



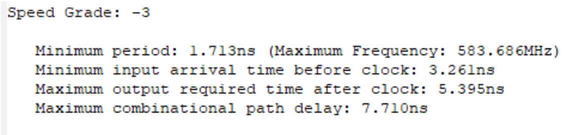
bwar Project Status (02/18/2020 - 09:49:41)			
Project File:	Error.xise	**Parser Errors:**	No Errors
Module Name:	bwar	**Implementation State:**	Synthesized
Target Device:	xc6slx9-3tqg144	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	13 Warnings (13 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	12	11440	0%
Number of Slice LUTs	15	5720	0%
Number of fully used LUT-FF pairs	8	19	42%
Number of bonded IOBs	28	102	27%
Number of BUFG/BUFGCTRLs	1	16	6%

TOP MODULE DESIGN:



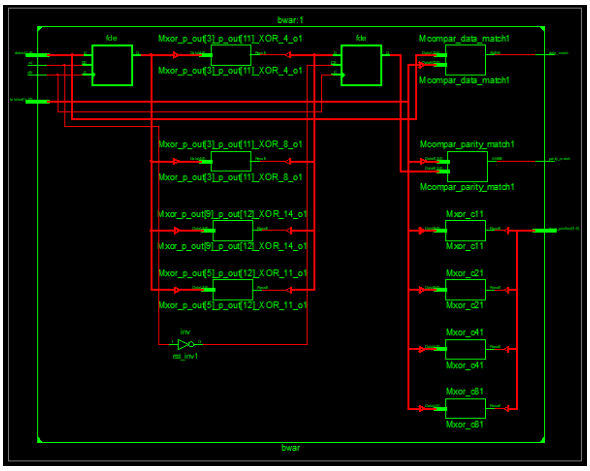
TIMING ANALYSIS:



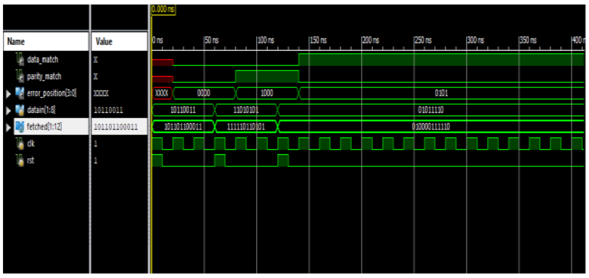
```
Speed Grade: -3

   Minimum period: 1.713ns (Maximum Frequency: 583.686MHz)
   Minimum input arrival time before clock: 3.261ns
   Maximum output required time after clock: 5.395ns
   Maximum combinational path delay: 7.710ns
```

RTL SCHEMATIC:


bwar:1
fde
Mxor_p_out[3]_p_out[11]_XOR_4_o1
Mxor_p_out[3]_p_out[11]_XOR_4_o1
fde
Mcompar_data_match1
Mcompar_data_match1
Mxor_p_out[3]_p_out[11]_XOR_8_o1
Mxor_p_out[3]_p_out[11]_XOR_8_o1
Mcompar_parity_match1
Mcompar_parity_match1
Mxor_p_out[9]_p_out[12]_XOR_14_o1
Mxor_p_out[9]_p_out[12]_XOR_14_o1
Mxor_c11
Mxor_c11
Mxor_p_out[5]_p_out[12]_XOR_11_o1
Mxor_p_out[5]_p_out[12]_XOR_11_o1
Mxor_c21
Mxor_c21
inv
rst_inv1
Mxor_c41
Mxor_c41
Mxor_c81
Mxor_c81
bwar

OUTPUT



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 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]. P. Kishore, P.V. Sridevi and K. Babulu "Design of Low voltage, Low Power and High-Speed Logic Gates Using Modified GDI Technique", International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 7, No. 2, pp.435-444,2016.

[2]. Pinninti Kishore, P.V. Sridevi and K. Babulu, "Low Power and Optimized Ripple Carry Adder and Carry Select Adder Using MOD-GDI Technique" Proceedings of ICMEET 2015, Lecture Notes in Electrical Engineering, pp 159-171, Springer India 2016.

[3] 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 Very Large-Scale Integration (VLSI) Systems, Vol. 22, No. 7, July 2014.

[4] J. Chang, M. huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen, S. Chiu, R.Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, Prinzie, "The 65-nm 16-MB shared on-die L3 cache for the dual-core Intel xeon processor 7100 series" IEEE J. Solid-State Circuits, vol. 42, no. 4, pp. 846-852, Apr.2007.

[4] J. D. Warok, Y.-H. Chan, S. M. Carey, H. Wen, P.J. Meaney, G. Gerwig, H. H. Smith, Y.H. Chan, J. Davis, P. Bunce, A. Pelella, D. Rodko, P. Patel, T. Starch, D. Malone, F. Malgioglio, J. Neves, D. L. Rude, and W.V. Huott "Circuit and Physical design implementation of the microprocessor chip for the zEnterprise System." IEEE J. Solid State Circuits, vol. 47, no. 1, pp. 151-163, Jan. 2012.

[6] 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.

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

[8] AMD Inc. (2010). Family 10th AMD Opteron Processor Product Data Sheet, Sunnyvale, CA, USA

[9] 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.

[10] M.Naga Kalyani and K.Priyanka, “Systematic Error-Correcting Codes Implementation for Matching of Data Encoded”, International Journal of Engineering Sciences & Research Technology, [Kalyani*, 4.(9): September, 2015.

[11] Mary Francy Joseph and Anith Mohan, “Improved Architecture for Tag Matching in Cache memory Coded with Error Correcting Codes”, Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016).

[12] Steffi Philip Mulamoottil1 Dr. E. Nagabhooshanam Nimmagadda Ravali, “Synthesis of Data Encoded with Error Correcting Codes Using BWA Architecture”, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834, pISSN: 2278-8735.Volume 10, Issue 4.