Low Complexity & Improved Efficiency of Encoded Data Using Peres Gate in BWAR with Testable Feature

Tripti Nirmalkar
Department of Electronics & Communication
Chhattisgarh Swami Vivekananda Technical University
Bhilai, India
e-mail: triptinirmalkar2904@gmail.com

Deepti Kanoujia
Department of Electronics & Communication
Sri Shankaracharya College of Engineering & Technology
Bhilai, India
e-mail: dk1325@gmail.com

Kshitiz Varma
Department of Electronics & Communication
Chhattisgarh Swami Vivekananda Technical University
Bhilai, India
e-mail: kshcsvtu@gmail.com

***Abstract*—In this present era of high speed developing world of VLSI, data comparison is broadly used in computing system. In a computation system the receiving data needed to compare with deposited information to trace the identical entry. Comparison of data is a technique which fetches data together from various sources and compares it. When this matching differ 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 study, it is found that the data matching can be done both in encoder and decoder end. The kept data preserved through error correcting codes (ECC code word) is used to relate with arriving data after decryption. 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. For a ECC code (16, 11) the proposed BWAR architecture minimizes the hardware complexity by 30.4% approximately with 13% of improved efficiency. In this brief, we have used a Xilinx 13.2 for Verilog coding for data matching algorithm.**

Keywords-Data comparison, Error Correcting Codes (ECC code word), Systematic Codes, Hamming Distance, Saturate Adder (SA), Butterfly Weight Accumulator (BWA), Reversible Logic Gates, Peres Gate, Peres Half Adder (HAP), Renovated Butterfly Weight Accumulator (BWAR).

I. INTRODUCTION

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 [1]. 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 effects 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 [2]-[6], 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 HAs 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.

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00029

II. EXISTING RESEARCH

In this section the existing architectures for data comparison are discussed. They are:

i. Conventional Method and Direct compare method
ii. Saturate adder (SA) based architecture
iii. Data path design for systematic codes and Hamming Distance computation architecture

Conventional Method: This is also called decode-and-compare architecture is used to recognize whether the data is match or mismatch. In Fig.1 the architecture of decode-and-compare architecture is shown where the cache memory with n-bit fetched code word is first decoded in the decoder for the recovery of the earliest k-bit data, meanwhile the original k-bit incoming code word is correlate in the comparator to check whether the source data and the fetched code word is match or mismatch. But in this mechanism, the hurdle is critical path, which is also extended to be utilized in realistic hideout structure for speedy approach. Resulting, the decoder is one other greatest complicated processing feature and takes extra period than encrypt, so, the complexity could not be negotiated.

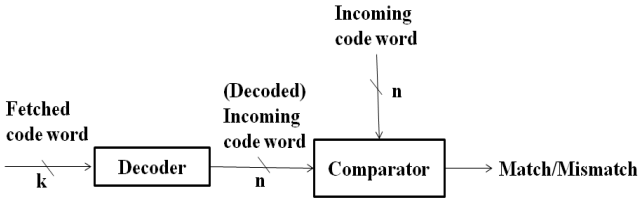


Fig. 1 Conventional Method [1]

To determine the drawback of the conventional method, the decoding of the save code term is reinstated thru the encoding of an arriving tag in direct link process.

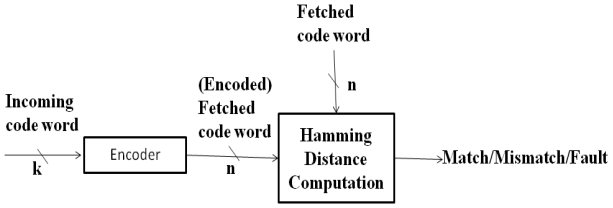


Fig. 2 Direct Compare Method [1]

Direct Compare Method: This architecture is a kind of direct compare method where encoded incoming code word is directly compared with the fetched code word [8].

A k-bit receiving links primarily encoded to equivalent n-bit fetched code word n(X) and distinguished with the fetched code word n(Y) as shown in Fig.2. This architecture has 3 steps:

a) Conceal the coming data.
b) Estimate the hamming distance
c) To examine the interval of t_{max} and r_{max}

Certify r_{max} and t_{max} signify the numbers of the largest detectable and correctable errors jointly. Given incoming information n and its encrypted data is showed as X. The fetched code word since memory is signified as Y. Therefore, the hamming distance d stands as [9];

d = distance(X, Y)

The folder is discussed as:

1) If $d = 0$, X meets Y accurately.

2) If $0 < d \leq t_{max}$, X will meet Y supply at maximum t_{max} flaws in Y are rectified.

3) Uncertainty $t_{max} < d \leq r_{max}$, Y has perceptible with unchanged errors.

4) If $r_{max} < d$, X does not meet Y.

Suppose arriving code word takes zero error, we can say, dual tags are matched, if d is in either the first or the second length. In this fashion, the architecture eliminates the decoder from its analytical track, maintaining the error correcting proficiency, at the new low cost of an encoder being proposed. We notice this, generally encode is even easier than the decoder, thus, the encoding expense automatically reduces, as compared to the decoding cost.

c) **Saturate Adder (SA) based Architecture:**

Decode and compare is a prolonged technique of data decoding and matching with complex architecture. Encode and compare is a direct method with less complexity along with the Hamming distance computation. This technique can detect the data mismatch and fault position using saturate adders (SA). Meanwhile, the overhead technique desires to calculate the Hamming distance; SA is defined as a primitive building section for computing the same.

The first step in current SA-based architecture is to perform XOR actions aimed at each combination of parts in X and Y. This generates a vector which represents the bit wise dissimilarity of both the code-words [10]. The Half Adders followed by this above discussed XOR gates, counts the total number of 1s in dual abutting bits in the track. The records of 1s remain gathered by traveling over the subsequent Saturate Adder stock [10]. In the SA stock, the accrued value z is drenched towards $r_{max} + 1$ uncertainty this one is higher than r_{max}. Also, given intakes x, y and z can be revealed as follows:

$$z = \begin{cases} x + y, & \text{if } x + y \leq 1 \\ r_{max} + 1, & \text{otherwise} \end{cases} \quad (1)$$

Lastly, the range of d is indicated by the final accumulated value.

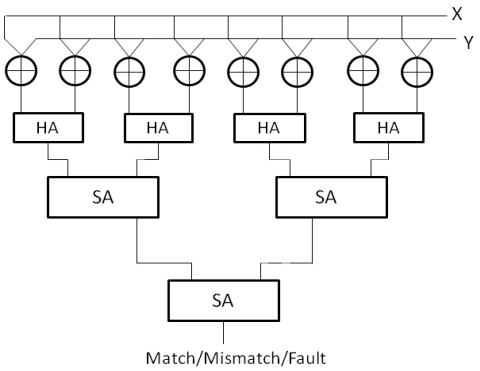


Fig. 3 Saturate Adder located architecture for the Encode-and- Compare Architecture [1]

But it is found that, because of this saturation, additional logic circuitry is any how needed, hence, the difficulty of a SA is greater than the common adder [11]. The critical path is also further lasting which makes an SA one additional sophisticated processing material. The proposed work replaces this complex SA with a renovated BWA reducing complexity and improved efficiency.

d) As compared to the above discussed existing works another new architecture was introduced for data comparison and hamming distance computation by using the typical feature of systematic codes. Now the view of this construction the encoding method to create the parity bits as of the received tag is executed in parallel with the tag comparison lowering the complete latency as exposed in Fig.4.

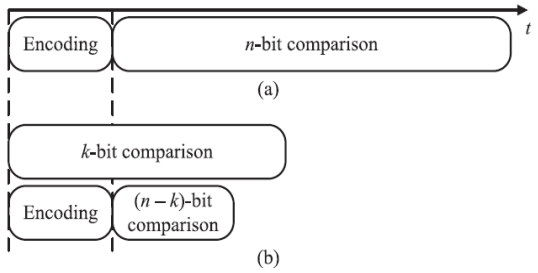
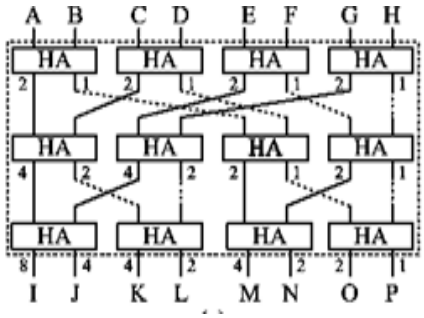


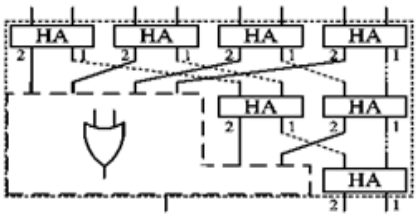
Fig. 4 Timing diagram of the data match in (a) encode-and-compare architecture [7] and (b) suggested style [1]

Furthermore, this data path design is found to be exists of numerous BWA to compute hamming distance with reduced complexity and delay. Basically this architecture counts the numeral of 1's amongst its input part that resides of many half adders. Every output part of a half adder related with a weight. The half adders in a level stay affiliated in a butterfly form to collect the transfer bits then the sum bits of the above stage distinctly [1]. For instance now in Fig.5 the highlighted Half Adder if the carry bit is decided then the number of 1's among the united input bits A, B, C and D is 2 and at the uttermost stage the number of 1's between the input bits, d, can be estimated as:

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



(a)



(b)

Fig. 5 Suggested BWA. (a) Common structure and (b) Novel structure updated for the comparing of ECC-protected information [1]

Possibly this simplified the circuit accurately. For example at $F_{SATMAX} = 1$, two or beyond two 1's between the input bits are considered the same lying in the fourth range and hence several HAs is replaced with easy OR-gate diagram. Thus, this architecture is preferred over SA that recourses the saturation. There is no flap among any couple of two carry-bit or sum-bit lines. In this it is easy to fixed flaps that offer multiple routing layers independent of the number of bits taken by the BWA.

III. PROPOSED ARCHITECTURE

In this section, an area efficient and improved efficiency new architecture is proposed for:

i) Comparison of data using the style of systematic codes (ECC code word).

ii) Architecture which calculate the Hamming Distances namely Renovated Butterfly-Weight-Accumulator (BWAR).

iii) Peres Gate Half Adder (HAP).

1) 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.4 where;
n= total number of bits (data part + parity part)
k= data part
(n-k) = parity part

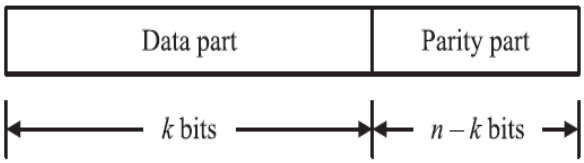


Fig. 6 Systematic representation of an ECC code word [1]

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 [12]. 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.

2) BWAR for Hamming Distance Estimation: The proposed architecture is described in Fig.7 which is grounded on the functional unit design [13]. A multiple BWARs are proposed for estimation of the hamming distance. As in proposed BWAR no additional logic are implem1ented, 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 [7]. 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.5. 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 [14]. The proposed architecture for the renovated Butterfly Weight Accumulator is drawn in the next Fig.

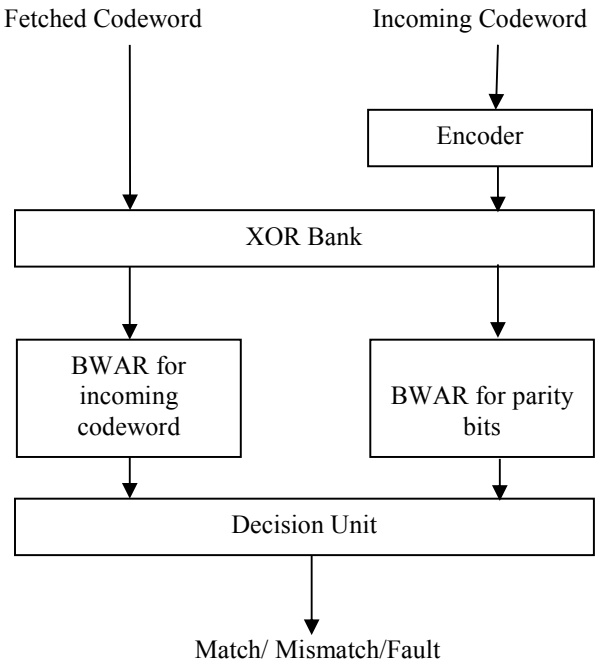


Fig. 7 Proposed BWAR based Architecture

3) Peres Half Adder (HAP) Approach: As we have studied earlier, half adder is one of the basic building blocks of BWAR, the main focus of this proposed work is to design a modified half adder using peres gate to reduce the complexity, power efficient and better performance architecture for data comparison. Peres gate is one of a kind of reversible logic gates which has attained great interest in the low power VLSI design.
Reversibility in calculating signifies that no data around the computational situations can endlessly be missing and thus the incoming tag can be recovered by computing towards the back or guesstimate result. This is termed as a logical reversibility [15].

3.1 Reversible Logic Gates: As discussed in previous paper, reversible logic gate is one-to-one plotting logic maneuver with n numbers of input and output logic which supports to describe the outputs after the inputs and likewise the inputs can be inimitably improved from the outputs which develop testable feature of whole architecture [15]. Thus, this logic best fits in our data comparison system. There are dissimilar types of basic alterable logic gates. The main advantage of this flexible circuit is that it must be planned with lowermost amount of reversible logic gates. The parameters defining the complexity and presentation of the circuit are:

- N = in a circuit the amount of gates used.
- CI = the quantity of continuous inputs, set towards either one 0 or 1 to produce the certain logical meaning.

- GO = garbage outputs i.e. unused outputs. (This cannot be ducked as these are very crucial to reach reversibility).
- QC = corresponding to the cost of a primitive gate the cost of the circuit. (This can be designed knowing the number of original reversible logic gates e.g. 1*1 or 2*2 needed to understand the circuit) [16].

In proposed architecture of BWAR according to the circuit requirement we use peres gate to design one section of BWAR comprising of half adders. These half adders are specially designed in this research using peres gate to reduce the complexity of the half adder resulting in better efficiency and low power architecture. The peres modified half adder (HAP) is discussed below along with the basic design and working of peres gate.

3.2 Peres Gate: The gate consist of the 3 inputs and 3 outputs i.e., 3*3 reversible gate taking inputs as (A, B, C) plotting to outputs as (P ⊕A, Q = A B⊕R = (A.B) C). Thus, the peres gate has quantum cost as 4 [17]. This gate is castoff for of its lowermost QC. In Fig.9 and Fig. 10 a 3*3 Basic Peres gate and internal route of peres gate is shown respectively.

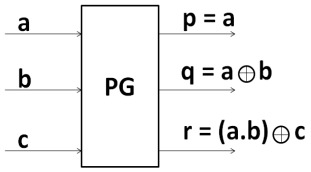


Fig. 9 Basic Peres Gate

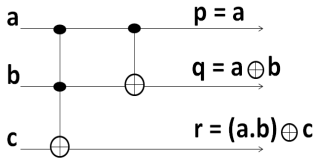


Fig. 10 Internal route of Peres gate

3.3 Exertion of Peres Gate in Half Adder:
The simple half adder consists of two 1-bit inputs and the output as sum and carry. The communal picture uses a XOR logic gate and an AND logic gate. The output for Half Adder is:

Sum = A ⊕ B
Carry = A & B

The Truth Table for half adder is assigned in TABLE I.

The Peres Half Adder is for the systematic codes, where the data bits and the parity bits are kept separated. Here we suppose a, b and c as input parts and p, q and r as output parts and design a half adder.

TABLE I. Truth table used for Half Adder

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

In peres half adder the input a assign to p and output q shows the (sum) as one output of half adder and r shows the (carry) as another output of half adder.
The Output of the Peres half adder is given as-
p = a;
q (sum) = a ⊕ b;
and
r (carry) = (a & b) ⊕ c.

The Fig.11 shows the Peres Half Adder and thr truth table is drawn in TABLE II.

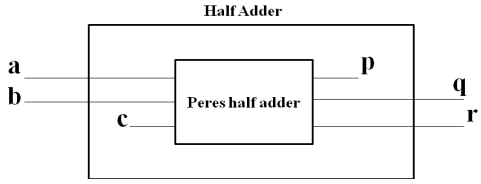


Fig. 11 Peres Half Adder

TABLE II. Truth Table of Peres gate

Inputs			Outputs		
A	B	c	p	q(sum)	r(carry)
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	1	0
1	1	0	1	0	1

IV. RESULT

Finally, the result drawn out of the entire research work modifying the half adder of BWA, the comparison unit, with peres gate was low complexity and improved efficiency, for an ECC code (16, 11).
As we know that, in the world of VLSI, mostly there is trade-off between area, speed and delay, in this proposed work, the human unnoticeable delay is found which increases in return to gain other aspects. The results are discussed in a tabular form in TABLE III.
The Proposed work is applied in two ways with the insertion of Peres Gate in Proposed Saturate Adder (SAP) and in the Renovated Butterfly Weight Accumulator (BWAR) from which we get the better result and is shown in the next table the reduction percentage of SAP and BWAR.

Table III. Comparison table of Existing and Proposed Work

Architecture (16, 11)	Complexity (Area)	Efficiency (A*D)	Delay (D)	Testable Feature
Existing Work for ECC using SA	132	1848	14ns	No
Existing Work for ECC using BWA	125	1500	12ns	No
Proposed Work for ECC using SAP	128	1664	13ns	Yes
Proposed Work for ECC using BWAR	87	1305	15ns	Yes
Reduction Percentage (%) Using SAP	3.03%	9.95%	-	-
Reduction Percentage (%) Using BWAR	30.4%	13%	-	-

The area (complexity) is found to be reduced by 30.4%. Efficiency is improved by 13%. As a trade-off between area, delay and efficiency in any VLSI design there is a human unnoticeable delay that can be negotiated because of improved complexity and area-delay product. It is found that peres gate contributes a lot in the improvement of the entire parameters and efficient data matching with simpler circuitry and testable feature.

V. CONCLUSION

In this proposed methodology, we have worked out for designing a new architecture inside the half adder of BWAR that is Peres gate which is one of the better type of a reversible logic gates which works efficiently to reduce complexity, even though the delay is human unnoticeable. Data comparison is done with the encoding process of parity bits and then compared with the data bits because data and parity bits should differentiated further. The efficiency of a proposed work is found to be improved. This is a reliable option for the comparison of the data protected with ECC. Even though, this work is only meant to match the incoming code word, still this can be utilized in applications that focus only on matching.

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] AMD Inc. (2010). *Family 10th AMD Opteron Processor Product Data Sheet,* Sunnyvale, CA, USA [Online]. Available: http://support.amd.com/us/Processor_Tech Docs/40036.

[7] 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.

[8] 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.

[9] 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).

[10] 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,p-ISSN: 2278-8735.Volume 10, Issue 4.

[11] D. Rajesh, D. Suneel, "Design of BWA based architecture For Matching Of Data Encoded With Error Correcting codes For Reduced Latency and Complexity", International journal of Science Techology & Management Vol. No.5, Issue No. 09, September 2016.

[12] Srinivas Aynapure and K S Vasundara Patil, **"**Hard Systematic Error-Correcting Codes for Matching of Data Using Low-Complexity Low-Latency Architecture", International Journal of Electrical ET & CSE Special Issue - NEWS 2016.

[13] Chaithanya K and Bhavya Das D, "An Area efficient Low Complexity Architecture for Comparing Data Encoded with Linear Block Codes", International journal of Advanced Research in Electrical, EIE, Vol. 5, Issue 6, June 2016.

[14] Priyanka, Wathodkar, Radhika R. Harne, "Architecture for Matching of Data Encoded with Hard Systematic Error Correcting Code using FPGA for Low Complexity and Low Latency" International Journal on Recent and Innovation Trends in C&C, ISSN: 2321-8169 Volume: 4 Issue: 5

[15] Prashant .R Yelekar, Prof. Sujata S. Chiwande, "Introduction to Reversible Logic Gates & its Application" 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA).

[16] Shefali Mamataj, Dibya Saha, Nahida Banu, "A Review of Reversible Gates and its Application in Logic Design" AJER e-ISSN : 2320-0847 p-ISSN : 2320-0936 Volume-03, Issue-04, pp-151-161.