International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.2, February 2015 DOI:10.15693/ijaist/2015.v4i2.59-63

Designing A Unique Architecture For Error Correcting Code To Reduce Complexity

S.Madurakali
PG Scholar, Department of VLSI Design
Sethu Institute of Technology, Pulloor,
Kariapatti,Virudhunagar.

Mr.T.Venishkumar M.E.,
Asst. Professor, Department of VLSI Design,
Sethu Institute of Technology,
Pulloor, Kariapatti,Virudhunagar.

***Abstract*- A new architecture for matching the data protected with an Error-Correcting Code (ECC) is presented in this project to reduce architecture 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 power consumption 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.To designing pipelined Vector Precoding architecture in decoder to reduce the error rate and hardware complexity.**

***Index Terms*-Butterfly-formed Weight Accumulator, Error Correcting Code, Vector Precoding,.**

I.INTRODUCTION

Butterfly-formed Weight Accumulator(BWA)is used to compute hamming distance between 2 code words such as X and Y.BWA is constructed by a set of HA.Data comparison is widely used in computing systems to perform many operations such as the tag matching in a cache memory and the virtual-to-physical address translation in a Translation Look aside Buffer (TLB).Hence, it is important to implement the comparison circuit with low hardware complexity. As recent computers employ Error-Correcting Codes (ECC) to protect data and improve reliability, complicated decoding procedure, which must precede the data comparison.

In performing the comparison, themethoddoes not examine whether the retrieved data is exactly the same as the incoming data.Instead, it checks if the retrieved data resides in the error correctable rangeof the codeword corresponding to the incoming data. As the checking necessitates an additional circuit to compute the Hamming distance, i.e., the number of different bits between the two code words.

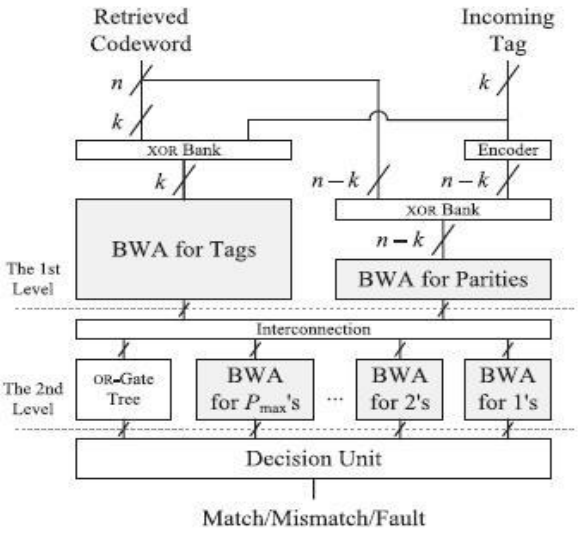

Figure. 1Proposed Architecture

Saturate adder (SA) is the key block in hamming distance computation unit.In this project, SA-based direct compare architecture is proposed to reduce the latency

]

and hardware complexity that computes the Hamming distance faster by resolving the aforementioned drawbacks.A saturation adder performs normal 4-bit addition when the resulting sum is less than 15. However, if the sum is greater than 15, which would normally cause an adder to roll over, the saturation adders output should stay at 15.

The rest of the paper is organised as follows: a brief basic concepts of BWA and handling methodologies are given in section II; Existing adder techniques are discussed in section III; the proposed innovative is propounded and manifested in section IV; the experimental results on simulation and the outcomes are discussed in section V; Finally in section VI conclusion of my work is elucidated.

II. BASICS

The most recent solution for the matching problem is the direct compare method, which 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.

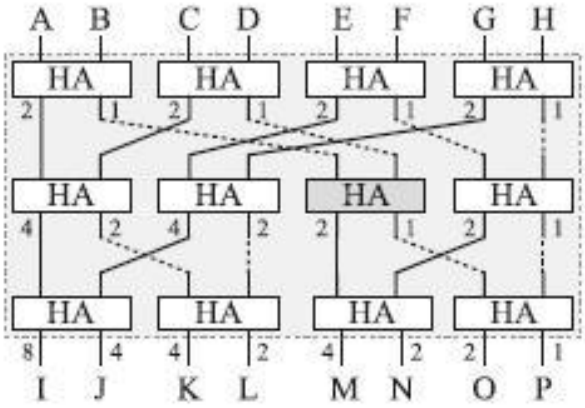

Figure. 2 Conventional BWA Architecture

Hamming Distance(d)=8*I+4(J+K+M)+2*(L+N+O)+P (1)

The matching or fault of the received code words are computed based on hamming distance computation method.Let R_{max}=maximum number of detectable errors in received code words.Let t_{max}=maximum number of correctable errors in received code words.

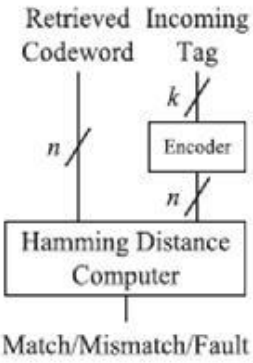

Figure. 3 Encode and Compare Architecture

Note that decoding is usually more complex and takes moretime than encoding as it encompasses a series of error detection or syndrome calculation, and error correction . To resolve the drawbacks of the decode-and-compare architecture, therefore, thedecoding of aretrieved codeword is replaced with the encoding of an incomingtag in the encode-and-compare architecture More precisely, a *k*-bitincoming tag is first encoded to the corresponding *n*-bit codeword *X*and compared with an *n*-bit retrieved codeword *Y* as shown inFig. 1(b). The comparison is to examine how many bits the twocodewords differ, not to check if the two codewords are exactly equalto each other. For this, we compute the Hamming distance *d* betweenthe two codewords and classify the cases according to the range of *d*.Let t_{max} and r_{max} denote the numbers of maximally correctable and detectable errors, respectively.

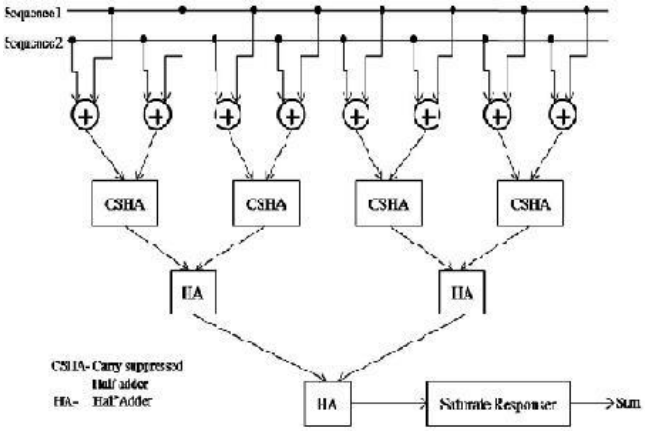

Figure. 4 SA-Based Comparator Architecture

In the SA-based architecture the comparison of two codewordsis invoked after the incoming tag is encoded. Therefore, the criticalpath consists of a series of the encoding and the *n*-bit comparisonas shown in Fig. 4. However, it does not consider the fact that, inpractice, the ECC codeword is of a systematic form in which thedataand parity parts are completely separated as shown in Fig. 4. As thedata part of a systematic codeword is exactly the same as the incomingtag field, it is immediately available for comparison while theparity part becomes available only after the encoding is completed.Grounded on this fact, the comparison of the *k*-bit tags can be startedbefore the remaining (*n–k)*-bit comparison of the parity bits. In theproposed architecture, therefore, the encoding process to generate theparity bits from the incoming tag is performed in parallel with the tag comparison, reducing the overall latency.

III. RELATED WORK

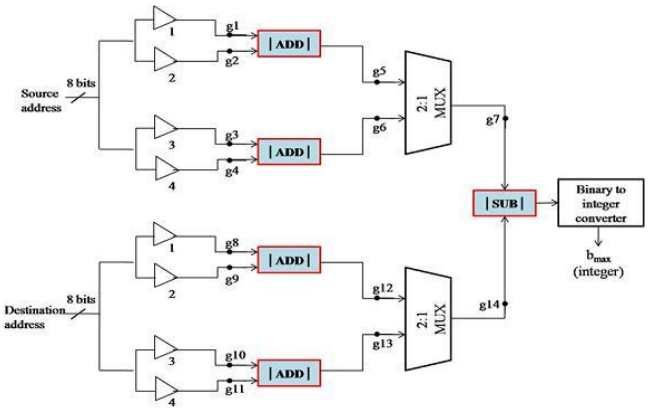

Figure. 5 Architecture of Encoder

In this,8 bits of source address and 8 bits of destination address are taken in to half-adder and to be add and be given in to mux. Finally that MUX value will be given to the subtractor. The subtracted value is in binary form. Using binary to integer converter, the binary bits will be converted in to integerform, that value denoted as $b_{max.}$

Nomenclature:	4-bit saturation adder
Data Input:	2, 4-bit vectors A, B
Data Output:	4-bit vector sum
Behavior:	if(A+B > 15) sum = 15) else sum = A + B

Figure 6 SA table

Q	R	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

Figure 7 Decision Table

Here data input is 2,4 bit vectors A,B. The data output is 4-bit vector sum and the resulting behaviour is greater than 15 or equal to 15 in the SA table.

In the decision table, six inputs are taken and they are labelled as Q, R, S, T, U, V. From the above decision table we can infer that, only when "0" bits are taken alone the decision is getting matched, where for the other combinations the results are either mismatched or fault.

IV. PROPOSED METHOD

The main objective is to design a new architecture in error correcting code to reduce hard ware complexity, latency and also to reduce burst errors in receiver side. In phase II, vector precoding architecture has been proposed to reduce noise in the receiving data which are from encoding the incoming data.

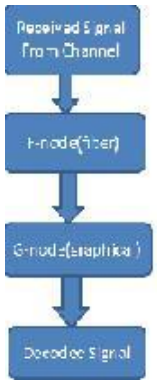

Figure. 7(a) Proposed vector precoding architecture

F node and g node are used to calculate the propagated LLR values. Here S2C is the block that performs the conversion from sign-magnitude form to 2's complement form, while C2S unit carries out the inverse conversion. Additionally, adder and subtractor are employed to carry out addition and subtraction between the two inputs. Finally, at the output end of the PE, *control* signal is used to determine the output as LLR(a) or LLR(b) ,which is propagated to the next stage.

If LLR (Log likelihood ratio) "c" is greater than LLR "d" then LLR "c" would be selected, If LLR "d" is greater than LLR "c" then LLR "d" would be selected. If LLR "c" is .
Equal LLR"d" then sum of LLR "c" and LLR "d" would be Selected.

Comparison	Selection
LLR(c)>LLR(d)	LLR(c)
LLR(d)>LLR(c)	LLR(d)
LLR(c)=LLR(d)	LLR(c) LLR(d)

34mW, and also the latency result is reduced to 15.139ms from 17.096ms

VI. CONCLUSION

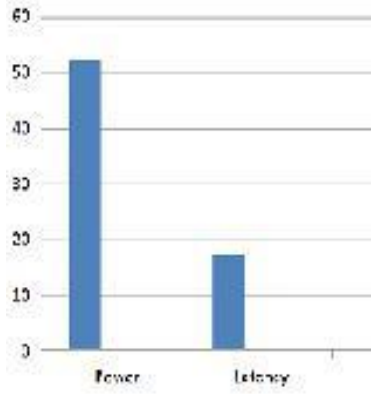

Figure. 8 Compare and select unit

V. RESULTS

The BWA Architecture is designed for error correcting codes also to reduced power consumption and latency as mentioned in the table.In phaseII vector precoding architecture is proposed in front of BWA architecture to reduced hardware complexion also reduced maximum utilization of power and latency as shown in the table.

Performance Analysis Parameter	Phase1 Results	Phase2 Results
Power Consumption	52mW	34mW
Latency	17.096ns	15.139ns

Figure. 9 Performance analysis

From the above table we can infer that the power consumption had been reduced in Phase 2 from 52mW to.

In this project, the low complexity architecture for error correction code is proposed.The Saturating adder based decoding architecture is designed and simulated in this review.The saturating adder is based on Half adders.The vector precoding architecture is designed for maximum reduction in power and latency.Here burst errors are reduced and minimum number of gates are used. Instead of using eight gates, here four gates are used

REFERENCES

[1] OEdfors, M Sandell, JJ Van de beek., "OFDM Channel estimation by singular value decomposition," 46th IEEE VVehicular Technology Conference Transactions, 1996.

[2] MPC Fossorier, M.Mihakujieve., "Reduced Complexity iterative decoding of low density parity Check codes based on belief propogation," IEEE Transactions, 1999.

[3]D Cassioli, MZ Win, Evatalaro., "Performance Of Low complexity RAK & reception in a Realistic UMB Channel., IEEE Transactions, 2002.

[4] H Topcuoglu, S.Hariri, M wu.,"Performance Effective and low complexity task scheduling for Heterogenous computing"., IEEE., 2002

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

[6]J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. Chen, W. Chen,"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.

[7] J. D. Warnock, Y.-H. Chan, S. M. Carey, H. Wen, P. J.
Meany, G. Gering 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.

[8].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.

[9]H.Khani., "Low complexity suboptimal monobit receiver for transmitted reference impulse radio UWB systems." IEEE Xplore, 2012.

[10] Y. Lee, H. You, and 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.

AUTHOR'SPROFILE

S.MADURAKALI received her B.Tech. degree in Electronics and Instrumentation Engineering from Kalasalingam University, Srivilliputhur, Virudhunagar, India, in 2012.Pursuing M.E.degreein VLSI Design from Sethu InstituteofTechnology,Anna University, Chennai, India. Her research interest includes digital electronics and VLSI design.

Dr.R.Ganesan received his B.E. in Instrumentation& Control Engineering from ArulmiguKalasalingam College Of Engineering and M.E.(Instrume- ntation) from Madras Institute of Tech- nology in the year 1991 and 1999 respectively. He has completed his Ph.D. from Anna University, Chennai, India in 2010. He is presently working as Professor and head in the department of M.E-VLSI Design at Sethu Institute of Technology, India. He has published more than 25 research papers in the National & International Journals/ Conferences. His research interests are VLSI design, Image Processing, Neural Networks and Genetic algorithms

.