https://doi.org/10.46335/IJIES.2023.8.3.10 e-ISSN: 2456-3463
Vol. 8, No. 3, 2023, PP. 52-57

International Journal of Innovations in Engineering and Science, www.ijies.net

An Implicit Multidimensional Parity based Data Coding and Decoding Algorithms with FPGA Implementation in Communication Systems

Mr. Vijay Santosh Tawar[1], Mr. Machchhindra Jibhau Garde[2]

[1]Assistant Professor – Department of Electronics & Telecommunication Engineering,
SSVPS' B. S. Deore College of Engineering, Dhule, India, 424002
[2]Assistant Professor – Department of Electronics Engineering
SSVPS' B. S. Deore College of Engineering, Dhule, India, 424002

kvtawar@gmail.com

***Received on**: 29 March, 2023* ***Revised on**: 17 April, 2023* ***Published on**: 19 April, 2023*

***Abstract –**The usability and use of devices in wireless communication is increasing gradually. The mobility and security of the system and the data integrity in the communication process are the requirements of the communication technology for the system. This is the main reason for the development of low voltage generators. Wireless communication medium consists of multiple signals, low voltage communication that can affect any system. Therefore, in wireless systems operating with low-power signals, it is best to use good data encoding and decoding strategies to send data over the medium.The data decoder can recognize changes in the data transmitted in the communication method and correct the data. This encoding-decoding process is calledDetection and Correction of error (EDAC) coding. One of such techniques, most often used in communication, is based on the concept of parity encoding and decoding. In the presented study, a multidimensional parity check based coding-decoding algorithm is proposed based on its results and simulation. With the help of proposed work, the associated process can be effectively used to detect and correct 3-bit errors in data transmission.*

***Keywords** -Encoder, Decoder, FPGA, HVD, Parity, EDAC, Xilinx.*

I. INTRODUCTION

In recent communication, wireless technology has emerged as part of the system. With the development of consumer products, data transmission is also increasing. The transfer of user data is also increasing due to free user-based applications. This provides greater flexibility for mobility and system upgrades than previous technology. Due to the many complex issues in the communication environment, wireless systems are more likely to be affected by noisy signals received by the antennas of the receiving system. Therefore, systems that handle the use of data acquisition signals must have fault detection procedures. Better methods than error detection methods also make corrections to the received data when errors are detected in the received data. Modification of received data can be controlled at the receiving end using the received data in addition to the original data from the sender before sending. However, where the perturbation of the input data is random in nature, it is difficult to choose the best estimate of the actual data. Many methods based on different algorithms have been proposed to solve this error. The main cause of the error is electromagnetic current in the communication channel or in the environment. This signal has the same properties, i.e.to. Noise caused by signal interference in real signal can be estimated. The amount of noise can be expressed as the number of transmission signals replaced by noise. Many researchers also refer to a small change as a single event upset (SEU) and a large change as a multiple bit upset (MBU). Since looking for interference in data is not only a good solution to problems in the erroneous systems, the system has also been developed to show correction of certain data affecting the bunch of bits. The implementation of this approach can be done at both the hardware and software level. While software systems are used to realize circuits in systems with limited cost, hardware-based circuits are preferred in systems that require higher performance. A simulated method in planning work involving error detection and error correction using four methods. The data is organized in the form of a matrix by creating parity bit streams in following the four matrix bit arrangements, such that - (i) horizontal, (ii) vertical, (iii) forward diagonal and (iv) backward diagonal arrangements. The generated parity bit-stream is written to the data frame at the appropriate time for transmission. The receiver circuits reproduce the parity bits, compare the received parity bits with the generated parity bits to check for errors, and use error correction hardware to complete its job.

II- LITERATURE REVIEW

The cheapest hardware is always preferred in communication devices with error detection and correction hardware. Few applications of detection of error and correction of error methods are multidimensional (H-V-D horizontal-vertical-diagonal) algorithms. The advantage of this algorithm is that bugs and fixes can be used effectively at both the hardware and software level. Previously, a simple technique to detect errors in a data set was to generate and check for parity. The parity bit can be an odd bit or even slightly different. A bit more of a bit balance only notices bit changes in the bit stream. With single-bit parity, any change in the data stream appears as correct data. This is the impetus for developing methods that use different bits or switch to other data encoding algorithms. An implementation of the multidimensional parity check based data handling scheme with multi-bit error detection and correction scheme in [1], [2], [3], [4] and [5]. A transient error-based detection and correction circuit that reduces the number of inputs and outputs in the connected circuits is shown in [6 and 7].A similar study is presented in [8] for satellite applications. In [9, 10 and 11], a software-decision-based approach using a 4 D parity based coding scheme is proposed. A 3 bit error detection and correction method is proposed in [12]. Flexible decision-based error correction codes for NAND flash are presented in [13]. H-V-D based fault protection is introduced in [14].An improved neighborhood error detection system for matrix-based codes is presented in [15]. Cache and memory error detection, correction and mitigation techniques for ground servers and workstations are presented in [16].

III- METHODOLOGY

The omnidirectional parity code proposed by Methodology is a coding method that has relationships in four directions, such as horizontal, vertical, forward, and backward diagonal. The parity design of the four directions is shown in Figure 1. In this way, the data is first arranged in matrix form. Matrix arrangement of long-length data is described in a particular way so that this arrangement can recognize four different direction based arrangement in the data block. Table 1 shows the data items in the block configuration of the matrix 8X8. Table 1 shows the parity items corresponding to the four different directions.H-V-D parity items are shown in Table 2. The proposed scheme has parity of 8 bits in the horizontal direction, 8 bits in the vertical direction, 15 bits in the diagonal in forward direction and 15 bits in the diagonal in backward direction. For a matrix of size "m x n", the total product in four directions is:

$[m + n + (m + n - 1) + (m + n - 1)] = 2[2(m + n) – 1]$.

The encoder input is hardware clock synchronized with the master reset input. The encoder functions to generate 4-odd parity data at the low logic reset control input.If the reset input is logic high, the encoder sets all internal registers and encoder outputs to logic low. The encoder has four equal parts connected together to produce the sameoutput. Each parity generator block is dedicated to generating parity for a particular direction. The functional diagram of the proposed encoder is shown in figure 2(a). The decoder input is a clock synchronous hardware design that is also a master reset control input.The operating diagram of the decision maker is shown in figure 2(b). The decoder operates on the received data and the equivalent bit of logic low resets the control input. If the reset input is logic high, the decoder sets all internal registers and the decoder's output to a logic without value. When enabled for processing, the decoder first generates horizontal, vertical, forward and backward diagonal parity bits from the received data bits. Thedecoder then compares the received parity bits with the generated parity bits. If the two parity sets are found to be unequal, the output error is set to a high value, otherwise a low value. If the error correction control input is enabled, the data is sent to the error correction logic block. If error correction causes the input to be set to logic low, no error correction causes data to be received. In this case, only information about the presence or absence of an error will be sent to the output of the previous block using the error indicator signal.

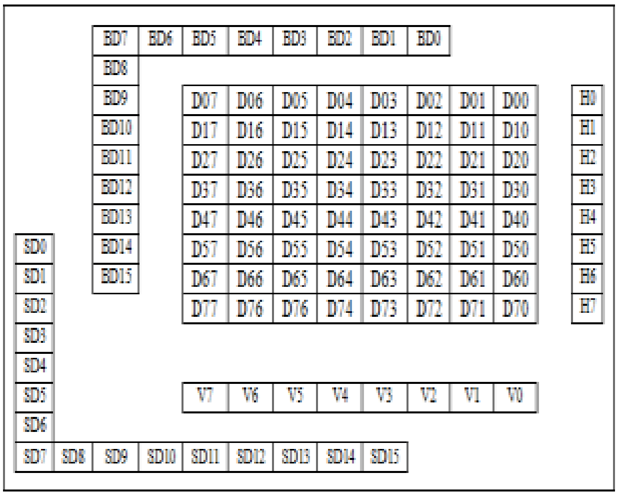

BD7 BD6 BD5 BD4 BD3 BD2 BD1 BD0
BD8
BD9 D07 D06 D05 D04 D03 D02 D01 D00 H0
BD10 D17 D16 D15 D14 D13 D12 D11 D10 H1
BD11 D27 D26 D25 D24 D23 D22 D21 D20 H2
BD12 D37 D36 D35 D34 D33 D32 D31 D30 H3
BD13 D47 D46 D45 D44 D43 D42 D41 D40 H4
SD0 BD14 D57 D56 D55 D54 D53 D52 D51 D50 H5
SD1 BD15 D67 D66 D65 D64 D63 D62 D61 D60 H6
SD2 D77 D76 D76 D74 D73 D72 D71 D70 H7
SD3
SD4
SD5 V7 V6 V5 V4 V3 V2 V1 V0
SD6
SD7 SD8 SD9 SD10 SD11 SD12 SD13 SD14 SD15

***Fig. 1:**Horizontal-Vertical-Diagonal H-V-D Parity based check method in 8X8 matrix form*

***Table 1:**Data blocks in 8X8 matrix form arranged in rows*

Row Number	Arranged Data Bits from MSB to LSB
1	D07-D06-D05-D04-D03-D02-D01-D00
2	D17-D16-D15-D14-D13-D12-D11-D10
3	D27-D26-D25-D24-D23-D22-D21-D20
4	D37-D36-D35-D34-D33-D32-D31-D30
5	D47-D46-D45-D44-D43-D42-D41-D40
6	D57-D56-D55-D54-D53-D52-D51-D50
7	D67-D66-D65-D64-D63-D62-D61-D60
8	D77-D76-D75-D74-D73-D72-D71-D70

***Table 2:**Generation of Parity Bits using Encoder*

Generationof parity bitsusing Encoder	
Horizontal Parity Bits	H0, H1, H2, H3, H4, H5, H6, H7
Vertical Parity Bits	V0, V1, V2, V3, V4, V5, V6, V7
Forward Diagonal Parity Bits	SD0, SD1, SD2, SD3, SD4, SD5, SD6, SD7, SD8, SD9, SD10, SD11, SD12, SD13, SD14, SD15
Backward Diagonal Parity Bits	BD0, BD1, BD2, BD3, BD4, BD5, BD6, BD7, BD8, BD9, BD10, BD11, BD12, BD13, BD14, BD15

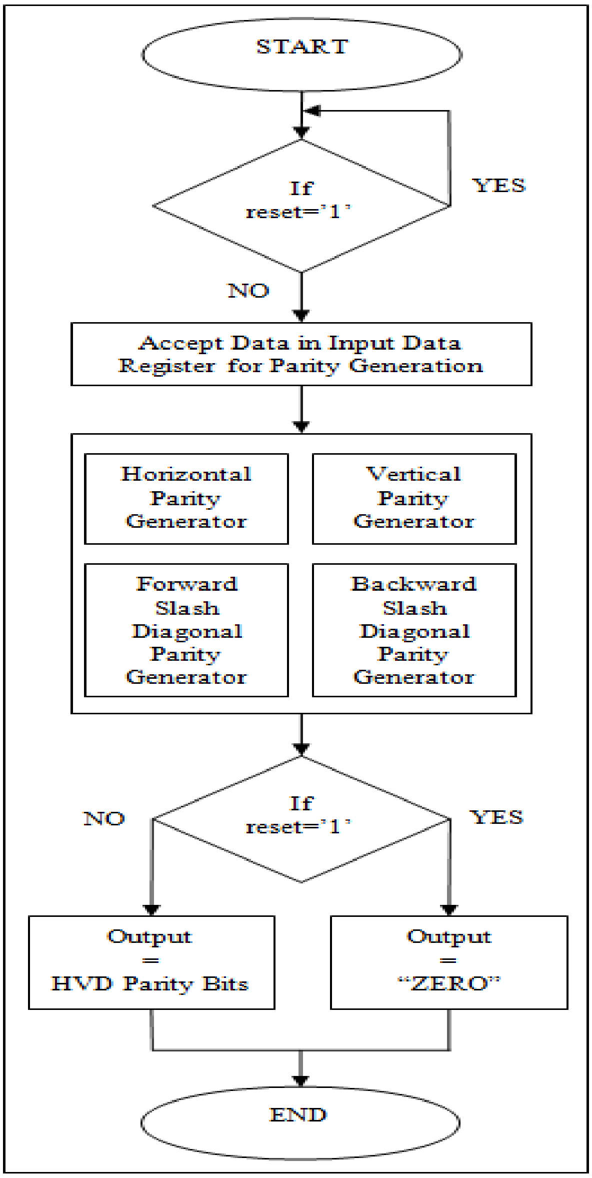

START
If reset='1'
YES
NO
Accept Data in Input Data Register for Parity Generation
Horizontal Parity Generator
Vertical Parity Generator
Forward Slash Diagonal Parity Generator
Backward Slash Diagonal Parity Generator
If reset='1'
NO
YES
Output = HVD Parity Bits
Output = "ZERO"
END

***Fig. 2:** (a) Encoder Block wise operation*

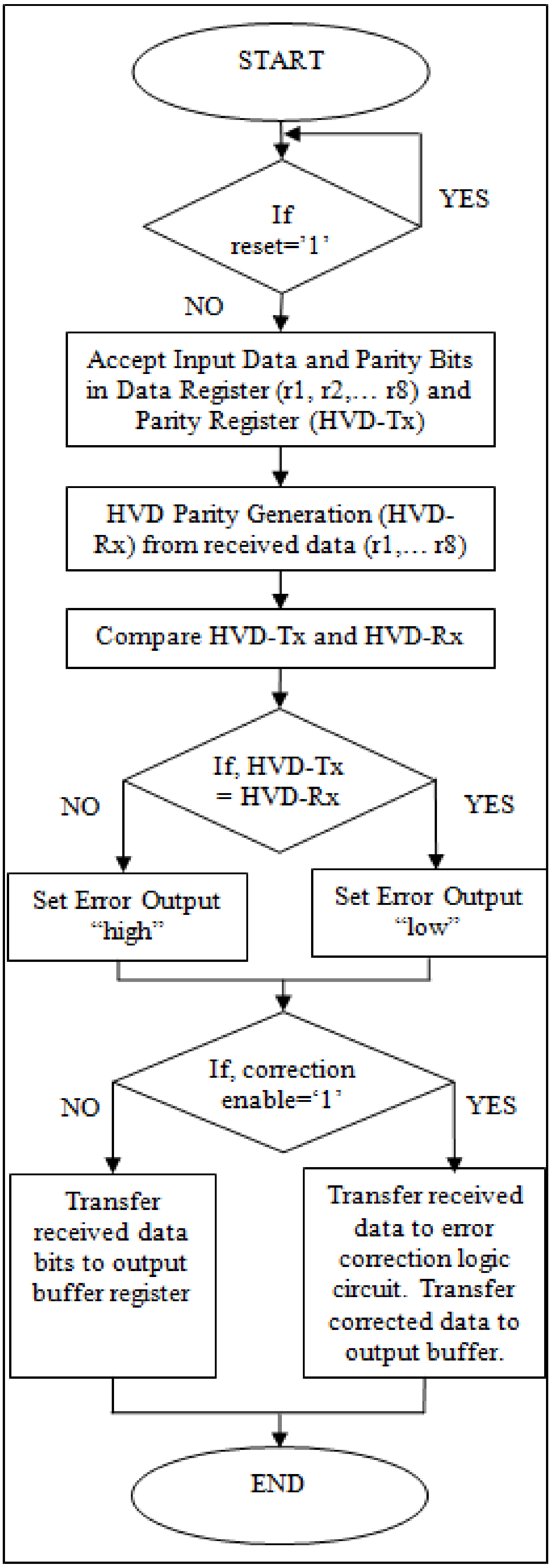

START
If
reset='1'
YES
NO
Accept Input Data and Parity Bits in Data Register (r1, r2,… r8) and Parity Register (HVD-Tx)
HVD Parity Generation (HVD-Rx) from received data (r1,… r8)
Compare HVD-Tx and HVD-Rx
If, HVD-Tx = HVD-Rx
NO
YES
Set Error Output "high"
Set Error Output "low"
If, correction enable='1'
NO
YES
Transfer received data bits to output buffer register
Transfer received data to error correction logic circuit. Transfer corrected data to output buffer.
END

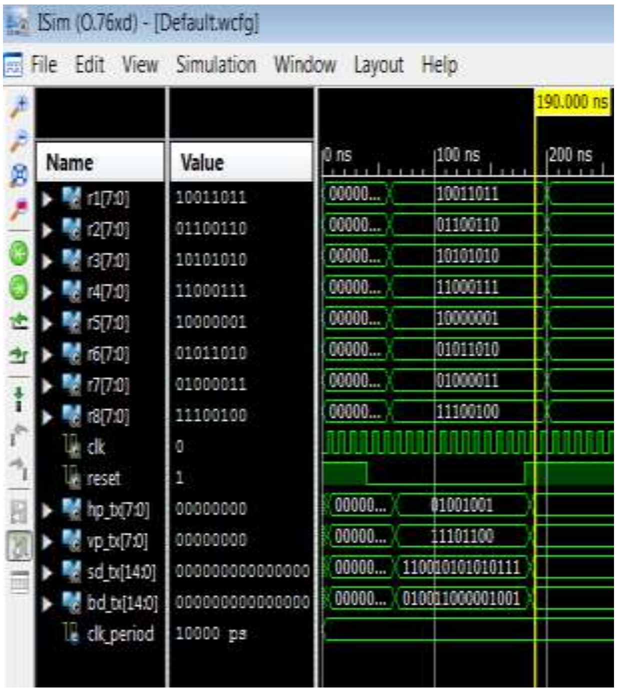
***Fig. 2:**(b) Decoder Block wise operation*

IV- SYNTHESIS&SIMULATIONOUTCOMES

The simulation of proposed H-V-Dtest bench design is done using Xilinx ISE Toolusing VHDL Language environment. The result showing waveforms of test bench simulation of encoder part is as shown in figure 3. The generated output parity bits along with binary data streams of input of receiver side are as follows –

Row No. 1 = 10011011
Row No. 2 = 01100110
Row No. 3 = 10101010
Row No. 4 = 11000111
Row No. 5 = 10000001
Row No. 6 = 01011010
Row No. 7 = 01000011
Row No. 8 = 11100100

Horizontal Parity Bits = 01001001
Vertical Parity Bits = 11101100
Forward Diagonal Parity Bits = 110010101010111
Backward Diagonal Parity Bits = 010011000001001



Name	Value
r1[7:0]	10011011
r2[7:0]	01100110
r3[7:0]	10101010
r4[7:0]	11000111
r5[7:0]	10000001
r6[7:0]	01011010
r7[7:0]	01000011
r8[7:0]	11100100
clk	0
reset	1
hp_tx[7:0]	00000000
vp_tx[7:0]	00000000
sd_tx[14:0]	000000000000000
bd_tx[14:0]	000000000000000
clk_period	10000 ps

***Fig. 3:**Simulated Output of Encoder*

The output waveforms of simulation of decoder are as shown in figures – 4 & 5. These waveforms are of following two cases:

Case I:when there is no error in the datareceived, and Case II:when 4-bit error is introduced and correction of erroneous datain the received data by the decoder. The inputs to decoder for simulation as given above two cases are as shown in following Table – 3:

Table 3: *Simulation Inputs of Proposed Decoder*

Inputs to Decoder	Case I	Case II
Row No. 1	10011011	10011011
Row No. 2	01100110	01100110
Row No. 3	10101010	10101010
Row No. 4	11000111	11000111
Row No. 5	10000001	10000001
Row No. 6	01011010	01011010
Row No. 7	01000011	01 **1111** 11
Row No. 8	11100100	11100100

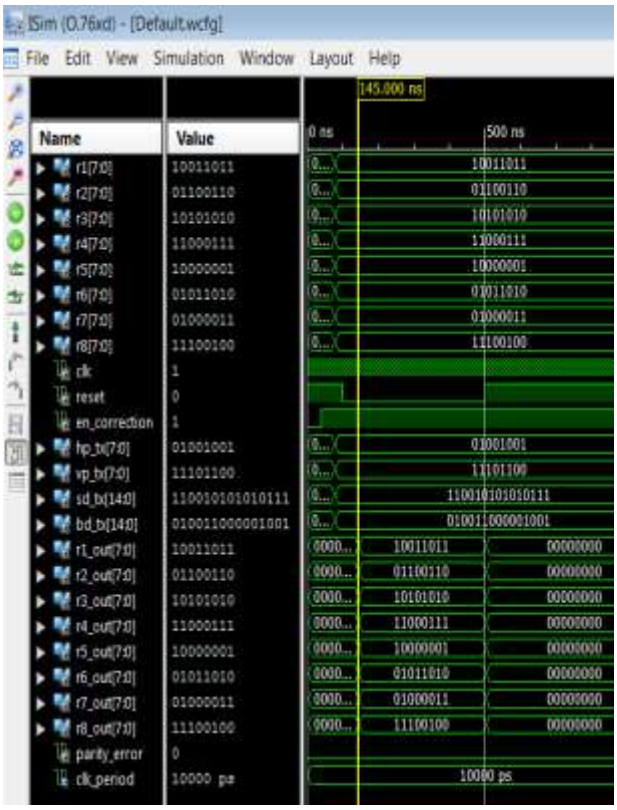
Xilinx Synthesis Tool on Xilinx Spartan 3E XC3S500E-4FG320 FPGA deviceplatform is used for performing synthesis of hardware based design proposed. Following tables 4 & 5 shows the optimum hardware utilization summary for proposed encoder & decoder designs.

Table 4: *Utilization summary of Hardware of proposed Encoder*

Hardware Resources used	Total	Encoder Utilization	
		Used	%
No. of Slices	4656	47	1
No. of LUTs	9312	68	1
No. of Flip-flops	9312	90	1

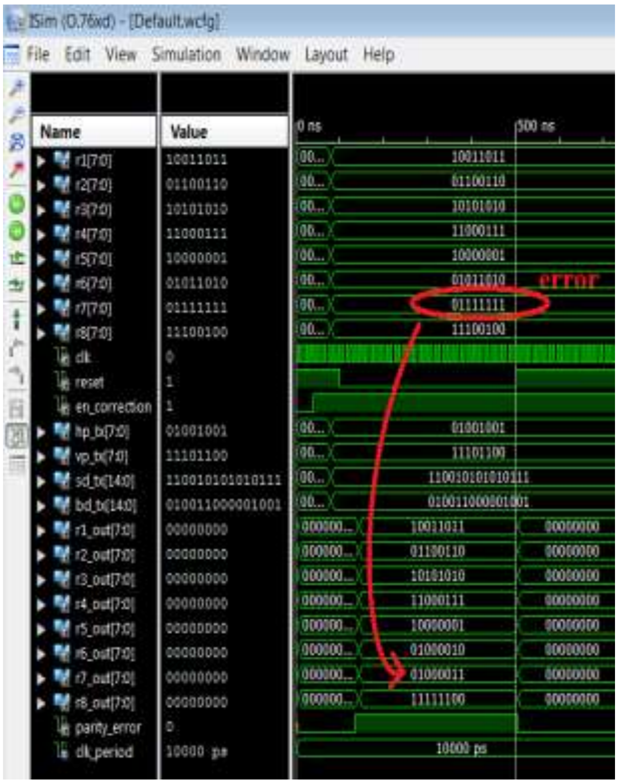
Table 5: *Utilization summary of Hardware of proposed Decoder*

Hardware Resources used	Total	Decoder Utilization	
		Used	%
No. of Slices	4656	227	4
No. of LUTs	9312	184	1
No. of Flip-flops	9312	329	3



Name	Value
r1[7:0]	10011011
r2[7:0]	01100110
r3[7:0]	10101010
r4[7:0]	11000111
r5[7:0]	10000001
r6[7:0]	01011010
r7[7:0]	01000011
r8[7:0]	11100100
clk	1
reset	0
en_correction	1
hp_tx[7:0]	01001001
vp_tx[7:0]	11101100
sd_tx[14:0]	110010101010111
bd_tx[14:0]	010011000001001
r1_out[7:0]	10011011
r2_out[7:0]	01100110
r3_out[7:0]	10101010
r4_out[7:0]	11000111
r5_out[7:0]	10000001
r6_out[7:0]	01011010
r7_out[7:0]	01000011
r8_out[7:0]	11100100
parity_error	0
clk_period	10000 ps

Fig. 4: *Simulated output for Case I of Decoder*



Name	Value
r1[7:0]	10011011
r2[7:0]	01100110
r3[7:0]	10101010
r4[7:0]	11000111
r5[7:0]	10000001
r6[7:0]	01011010
r7[7:0]	01111111
r8[7:0]	11100100
clk	0
reset	1
en_correction	1
hp_tx[7:0]	01001001
vp_tx[7:0]	11101100
sd_tx[14:0]	110010101010111
bd_tx[14:0]	010011000001001
r1_out[7:0]	00000000
r2_out[7:0]	00000000
r3_out[7:0]	00000000
r4_out[7:0]	00000000
r5_out[7:0]	00000000
r6_out[7:0]	00000000
r7_out[7:0]	00000000
r8_out[7:0]	00000000
parity_error	0
clk_period	10000 ps

Fig. 5: *Simulated output for Case II of Decoder*

V-RESULT & DISCUSSION

In this method of work, a technique for error detection and correction is proposed. The system design examines and corrects all sequential and multiple errors from data blocks less than 4 bits in length up to 3 bits. In exceptional cases the 4 bit error is also within the repair capacity of this algorithm. The execution strategy also generates output signals with error signals showing indications of errors. This output stimulates the interface circuitry to take decision whether to accept or reject the data received.

VI- CONCLUSION

Omnidirectional parity check algorithm plays an important role in detection and correction of errors in data communication, so the proposed work using the FPGA algorithm shows that the data capacity is improved and optimum hardware resources are utilized. The proposed algorithm can be improved in the future to correct all 3 bit and up to 4 bit errors. Additionally, the proposed operation can be improved by running simulations to detect and correct burst errors. This working concept can also be combined with other techniques to identify errors and correct the process as an attempt to improve the system.

REFERENCES

[1] *V Tawar & R Gupta, "A 4 Dimensional Parity based Data Decoding Scheme for EDAC in Communication Systems", IJRASET, April 2015.*

[2] *V Tawar & R Gupta, "FPGA Implementation of 4D Parity based Data Coding Technique", IJRET, March 2015.*

[3] *T. EtzionandE. Yaakobi, "High dimensional error-correcting codes", IEEE International Symposium on proceedings in Information Theory, 2010.*

[4] *M. Rubinoff, "N-dimensional codes for detecting and correcting multiple errors", Communications of the ACM, Volume-4, pp. 545-551, 1961.*

[5] *J.M. SheaandT. F. Wong, "Multi-dimensional parity check codes for bursty channels", IEEE International Symposium on proceedings in Information Theory, 2001.*

[6] *Luigi Carro, Fernanda Lima, Ricardo Reis, "Designing Fault Tolerant Systems into SRAM-based FPGAs", DAC Anaheim, California, June 2-6, 2003.*

[7] *D K Pradhan,C. Argyrides, H. R. Zarndi, "Multiple upsets tolerance in SRAM Memory", International Symposium on Circuits and Systems, New Orleans, 2007.*

[8] *Y. Bentoutou, "Program Memories Error Detection and Correction On-board Earth Observation Satellites", World Academy of Science, Engineering and Technology, 2010.*

[9] *M. Kishani, H. Pedram, H. Zarandi, A. Tajary, M. Raji and B. Ghavami, "HVD: Horizontal-Vertical-Diagonal error detecting and correcting code to protect against the soft errors", Design Automation for Embedded Systems, Volume-15, No 3-4, 2011.*

[10] *Z. Al-Ars,M. Imran and G. N. Gaydadjiev, "Improving Soft Error correction capability of 4-Dimensional parity codes", 14th IEEE European Test Symposium, 2009.*

[11] *S. Sharma and P. Vijayakumar, "An hvd based error detection and correction of soft errors in semiconductor memories used for space applications", International conference on Proceedings in Devices, Circuits and Systems (ICDCS), 2012.*

[12] *U. Thirunavukkarasu,N.B. Anne and S. Latifi, "Three and four-dimensional parity-check codes for correction and detection of multiple errors", International Conference on Proceedings in Information Technology: Coding and Computing (ITCC), 2004.*

[13] *Ningde Xie,Guiqiang Dong and Tong Zhang, "Enabling NAND Flash Memory use Soft-Decision Error Correction Codes at Minimal Read Latency Overhead", IEEE Transactions on Circuits and Systems, Vol. 60, September 2013.*

[14] *Hossein Pedram,Mostafa Kishani, Alireza Tajary,Hamid R. Zarandi, Mohsen Raji and Behnam Ghavami, "HVD: Horizontal-Vertical-Diagonal Error Detecting and correcting code to protect against soft errors", April 2011.*

[15] *C. Argyrides, P. Reviriego, D. Pradhan and J Maestro, "Matrix-based Codes for Adjacent Error Correction", IEEE Transaction on Nuclear Science, Vol. 4, August 2010.*

[16] *W. Slayman&A Charles, "Cache and Memory Error Detection, Correction and Reduction Techniques for Terrestrial Servers and Workstations", IEEE Transactions on Device and Material Reliability, Vol 5, September 2005*