FPGA Implementation of Multi Error Correction and Detection for IoT based WSN Applications

Jyoti Mani[1], Prof. Ashish Raghuwanshi[2]
[1]M.Tech Scholar, Department of Electronics and Communication Engineering
[2]Assistant Professor, Department of Electronics and Communication Engineering,
[1&2]IES College of Technology, Bhopal, India.

Abstract : Wireless communication network must be able to transfer data with high accuracy and reliability for various application. So to retain the more reliability of these networks, errors must be detected and corrected with great efficiency. There are number of error detecting and correcting codes available but most of them confront lot of challenges such as unreliable wireless links, the broadcast nature of wireless transmissions, interference, frequent topology changes, and other effects of wireless channels. This work proposed an implementation of multi error correction and detection using verilog code on xilnx 14.7 software. Proposed MEC-MED circuit design for high speed and less complexity and reduced area logic circuit. The conventional SEC-SED logic is modified to extend error correction and detection and improve latency, area, power, throughput and frequency. Simulated results shows that proposed MEC-MED VLSI architecture gives significant improved results than conventional SEC-SED.

IndexTerms – SEC, SED, MEC, MED, IOT, WSN, Verilog, VLSI, Xilinx.

I. INTRODUCTION

All error-detection and correction schemes add some redundancy (i.e., some extra data) to a message, which receivers can use to check consistency of the delivered message, and to recover data that has been determined to be corrupted. Error-detection and correction schemes can be either systematic or non-systematic. In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only error detection is required, a receiver can simply apply the same algorithm to the received data bits and compare its output with the received check bits; if the values do not match, an error has occurred at some point during the transmission. In a system that uses a non-systematic code, the original message is transformed into an encoded message carrying the same information and that has at least as many bits as the original message.

Figure 1.1: Basic Communication

Good error control performance requires the scheme to be selected based on the characteristics of the communication channel. Common channel models include memory less models where errors occur randomly and with a certain probability, and dynamic models where errors occur primarily in bursts. Consequently, error-detecting and correcting codes can be generally distinguished between random-error-detecting/correcting and burst-error-detecting/correcting. Some codes can also be suitable for a mixture of random errors and burst errors.

Convolutional codes are processed on a bit-by-bit basis. They are particularly suitable for implementation in hardware, and the Viterbi decoder allows optimal decoding.

Block codes are processed on a block-by-block basis. Early examples of block codes are repetition codes, Hamming codes and multidimensional parity-check codes. They were followed by a number of efficient codes, Reed–Solomon codes being the most notable due to their current widespread use. Turbo codes and low-density parity-check codes (LDPC) are relatively new constructions that can provide almost optimal efficiency.

Shannon's theorem is an important theorem in forward error correction, and describes the maximum information rate at which reliable communication is possible over a channel that has a certain error probability or signal-to-noise ratio (SNR). This strict upper limit is expressed in terms of the channel capacity. More specifically, the theorem says that there exist codes such that with increasing encoding length the probability of error on a discrete memoryless channel can be made arbitrarily small, provided that the code rate is smaller than the channel capacity. The code rate is defined as the fraction k/n of k source symbols and n encoded symbols.

The actual maximum code rate allowed depends on the error-correcting code used, and may be lower. This is because Shannon's proof was only of existential nature, and did not show how to construct codes which are both optimal and have efficient encoding and decoding algorithms.

II. LITERATURE OVERVIEW

X. Peng et al.,[1] This work introduces a solution for ultra-low bit-error-rate interface of superconductor-semiconductor. By using an error-correction-code encoder, the low bit-error-rate requirement could be relaxed considerably. The encoder in our research encodes 32-bit original data with six extra check bits that are generated based on the theory of linear block code to correct 1-bit error. The numerical analysis shows the bias margin of BER <; 10 -10 expands more than 50% for a reported typical interface. Besides, the mean time between failure of the 32-bit interface running at 5 Gbps with 10 -12 BER improves from 6.25 s to 44 000 years, and the BER requirement for 1 h operation without error is relaxed from 1.7 × 10 -15 to 1.14 × 10 -8.

S. Tripathi et al.,[2] The internet of things (IoT) based wireless sensor networks (WSNs) have been employed in many applications in industrial and medical fields. Smaller codeword length and low area complexity channel coding schemes are generally chosen in these applications. Delay and power consumption are the major issues in IoT related applications. In this work, a fast and power efficient single error correction-double error detection (SEC-DED) and single error correction-double error detection-double adjacent error correction (SEC-DED-DAEC) codes are proposed.

S. G et al.,[3] Here various error detection and correction methods are implemented based on parity check code along with hamming code algorithm. Encoder block may calculate the parity bits for data bits in various parity check code methods and also parity bits are coded with hamming codes. Whereas in decoder block, data can be regenerated after calculating syndrome vector and parity checking also takes place. Such encoder and decoder block is developed using verilog coding in Xilinx ISE software. Finally the comparison of various parity check code with hamming code is distinguished.

A. Zeh et al.,[4] We propose two parity-based concurrent error detection schemes for the Quarterround of the ChaCha stream cipher to protect from transient and permanent faults. They offer a trade-off between implementation overhead and error coverage. The second approach can detect any odd-weight error on the in-/output and intermediate signals of a Quarterround, while the first one requires less logic.

C. Shao et al.,[5] The nanoscale CMOS technology has encountered severe reliability issues, especially in the on-chip memory. As the technology nodes keep shrinking, single-event upsets (SEUs) may encounter more frequent multiple-bit upsets (MBUs) per particle strike. The commonly used memory error correction methods, such as the single-error correction-double-error detection (SEC-DED) code, are no longer feasible. While the counterparts for multiple error correction codes (MECs) yield too costly overhead on delay and data redundancy, especially for MBUs in a word or a character, even with all bits upset, the error correcting ability of the existing error correcting methods is exceeded. In this work, we introduce a novel block-based error detection and correction method for memory by analyzing the similarity of data. This method of error location and correction can cope with both single-word error (SWE) and multiple-word error (MWE), no matter how many corrupted bits of each word there are.

J. Gracia-Morán et al.,[6] A common solution is the use of error correction codes (ECCs). Nevertheless, when using ECCs in space applications, they must achieve a good balance between error coverage and redundancy, and their encoding/decoding circuits must be efficient in terms of area, power, and delay. Different codes have been proposed to tolerate MCUs. For instance, Matrix codes use Hamming codes and parity checks in a bi-dimensional layout to correct and detect some patterns of MCUs. Recently presented, column-line-code (CLC) has been designed to tolerate MCUs in space applications. CLC is a modified Matrix code, based on extended Hamming codes and parity checks. Nevertheless, a common property of these codes is the high redundancy introduced.

Y. Zhang et al.,[7] This work presents iRazor, a lightweight error detection and correction approach, to suppress the cycle time margin that is traditionally added to very large scale integration systems to tolerate process, voltage, and temperature variations. iRazor is based on a novel current-based detector, which is embedded in flip-flops on potentially critical paths. The proposed iRazor flip-flop requires only three additional transistors, yielding only 4.3% area penalty over a standard D flip-flop. The proposed scheme is implemented in an ARM Cortex-R4 microprocessor in 40 nm through an automated iRazor flip-flop insertion flow.

R. G. Rizzo et al.,[8] This work introduces Early Bird Sampling (EBS), a Razor variant that applies to low-power sequential circuits. The EBS allows to (i) solve the problem of short-path races bypassing tedious holdtime fixing design stages, (ii) reduce design overhead exploiting a local logic-masking mechanism for error correction. As a key feature, EBS enables Data-Driven Voltage Over-Scaling (DD-VOS), an aggressive dynamic voltage scaling strategy particularly suited for ultra-low power error-resilient applications. Simulation runs on a representative set of circuits provide a fair comparison with a standard Razor strategy. The collected results show EBS reduces area overheads (3.6% against 71.6% for Razor) and improves the voltage scaling profile achieving lower energy-per-operation (savings w.r.t. Razor range from 19.1% to 53.1%).

N. Sayed et al.,[9] The use of robust Error Correction Coding (ECC) with multiple bit correction capability to optimize the write margin and reliability results in large decoding latencies and large number of check bits. In this work, we propose a new solution to exclude the high costs of using ECC in terms of decoding latency and storage overhead to be able to use STT-MRAM for fast caches. We exploit the fact that STT-MRAM poses asymmetric errors due to the nature of Magnetic Tunnel Junction (MTJ) cell, which makes ECC a pessimistic solution to address such errors. Therefore, efficient Systematic Unidirectional Error-Detecting Code (SEDC) is proposed to be adopted instead of conventional ECC combined with proper cache access mechanism to fetch correct data in case of error detection. Our proposed approach provides orders of magnitude better reliability and considerable performance improvement that makes STT-MRAM viable for fast-caches compared to the existing solutions.

W. Jin et al.,[10] In this work, we analyze the hardware overhead of error detection techniques in pipelines based on three different sequential elements: flip-flops, two-phase latches, and pulsed latches. By exploiting the cycle-borrowing ability, we propose a technique called sparse insertion of error detecting registers on the two-phase latch-based and pulsed-latch-based pipelines to reduce the sequential logic area. Furthermore, we propose a delay-padding methodology using a multi-V_t cell library in ULV circuits to reduce EDAC hardware overhead. The proposed techniques are applied on a benchmark six-stage pipeline operating at 0.35 V in a 65-nm CMOS. The analysis results show that our proposed techniques can reduce the total area by 26%-33% and the error detecting register count by 2.9-4.3× compared with conventional EDAC techniques.

III. PROPOSED METHODOLOGY

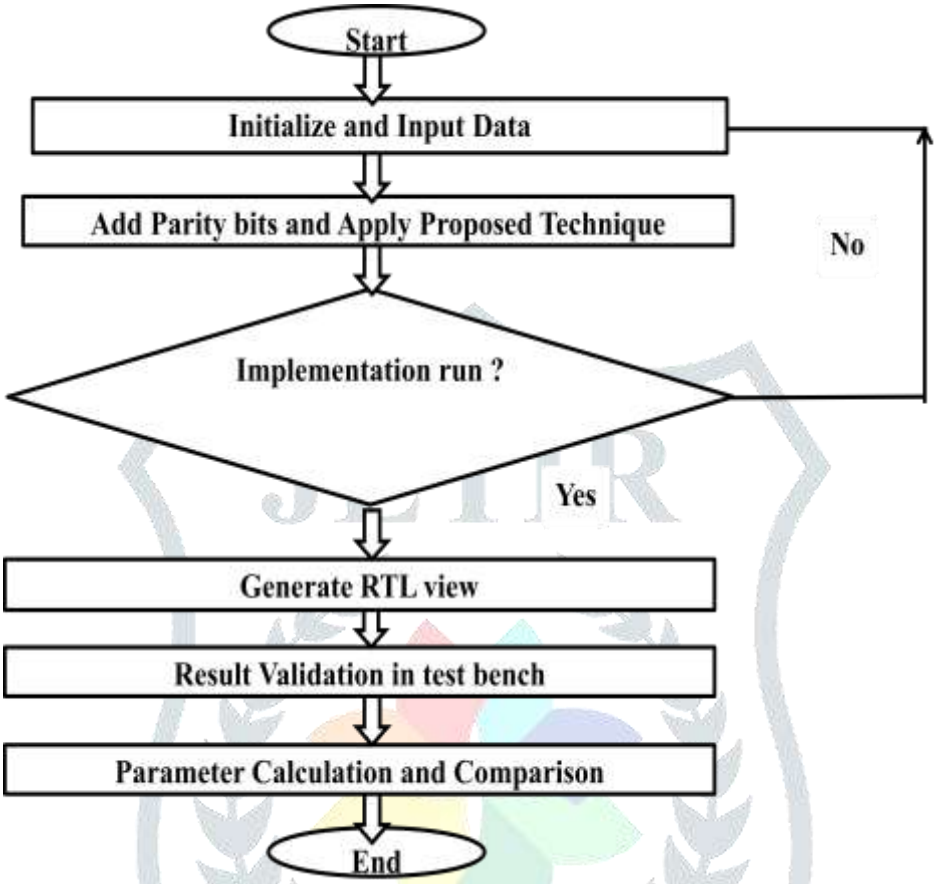

Figure 2: Flow Chart

Modified Single Error Correction - Double Error Detection

The H-matrix for proposed multi error correction- multi error detection (MEC-MED) codes have been constructed using following basic conditions:

1. All the columns are nonzero and distinct.
2. All data columns must have weight (w) three.
3. The XOR sum of any two columns should not be equal to any of the individual column.
4. The XOR sum of any two adjacent columns must be distinct and nonzero. The condition 1 is necessary for the mingle error correction (MEC) process. Double error detection property is confirmed by conditions 1, 2, and 3.

The procedure to generate the proposed H-matrix for MEC-MED codes is as follows:

Step 1: Initializing the H-matrix with (n−k) number of rows and (n) numbers of columns.

Step 2: The H-matrix consists of (n − k) numbers of parity columns having identity property and k numbers of data columns.

Step 3: Perform the modulo-2 operation between first parity column (p1) and last data column (d8).

Step 4: Perform the modulo-2 operation between last data column (d8) and its previous data column (d7).

Step 5: Continuing modulo-2 operation till second data column (d2) and first data column (d1) is performed.

Step 6: Place with '1' in data column's position which obtain from modulo-2 operations and remaining position with '0'.

IV. IMPLEMENTATION RESULT

The implementation and simulation of the proposed algorithm is done over Xilinx 14.7. The behavioral modeling style and Isim simulator is adopted for simulation. RTL and synthesis results are also generated.

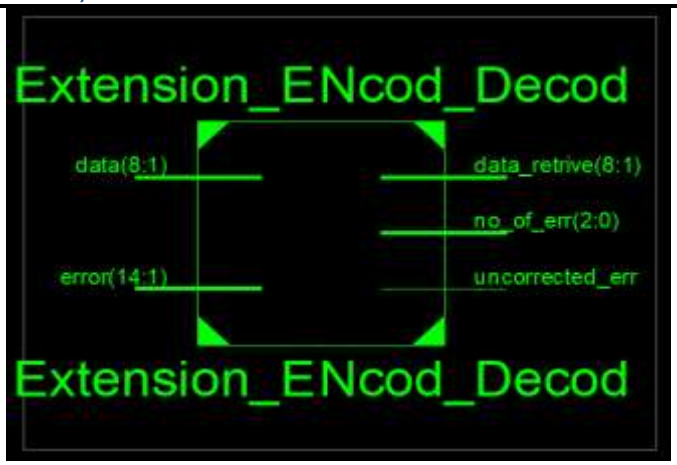

Figure 3: Top level module

Figure 3 is showing the top level module. Here the input is 8bit and apply the error upto 15 bit. The output side of block is 8 bit data retrieval (Recovered original data), number of error is 2 and 1 bit uncorrected error status.

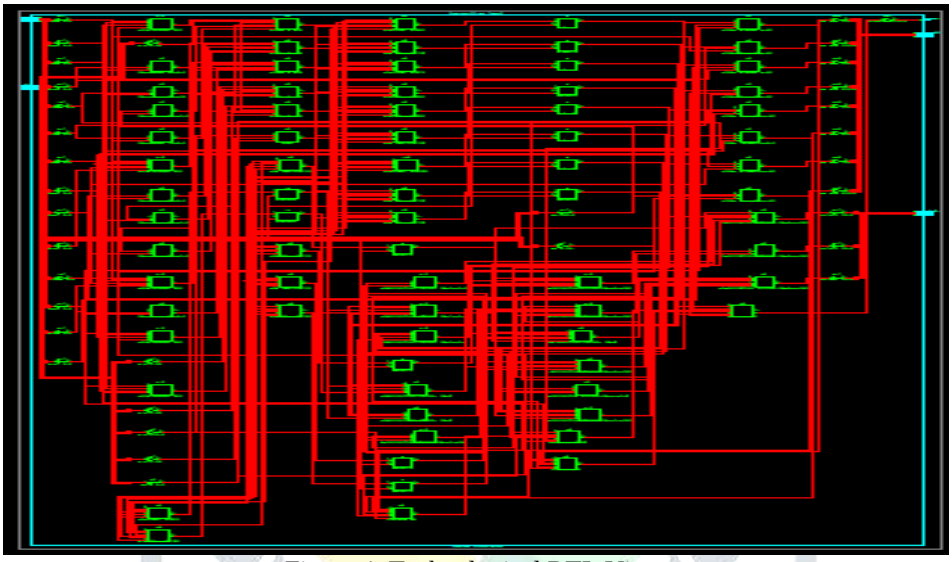

Figure 4: Technological RTL View

Figure 4 is providing the complete technological RTL view of proposed circuits.

Table 1: Device utilization summary

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	14	408000	0%
Number of Slice LUTs	61	204000	0%
Number of fully used LUT-FF pairs	14	61	22%
Number of bonded IOBs	34	600	5%

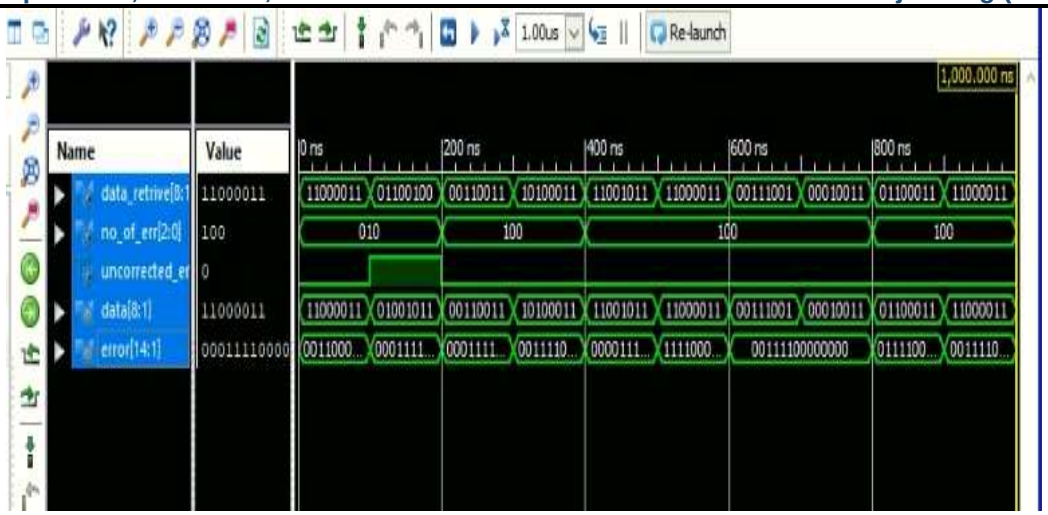

Figure 5: Data input with no error

Figure 5 is providing different input bits and out bits according the error and no error condition. Unexpected error shows bit 1 and signal shows at high conditions.

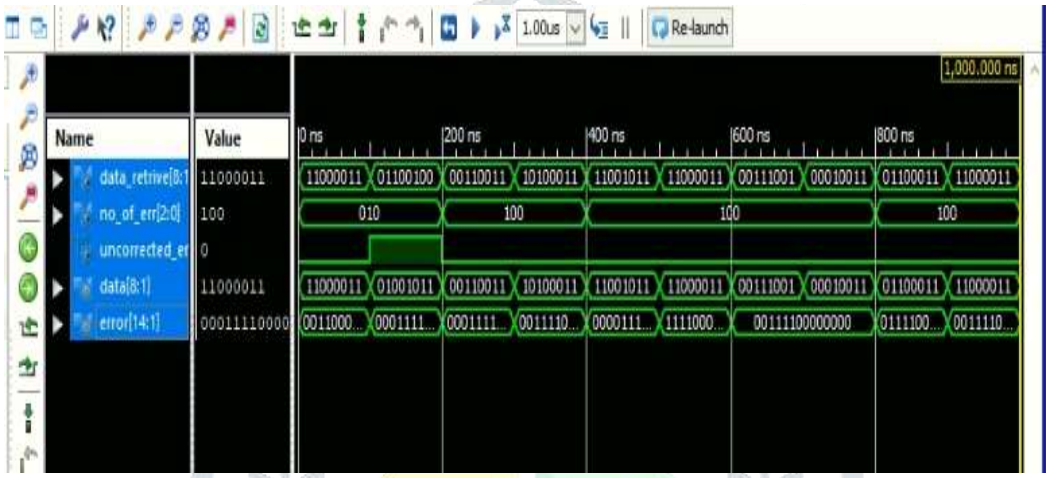

Figure 6: Error Transition zero

Figure 6 is providing different input bits and out bits according the error and no error condition. Error transition is zero in this case.

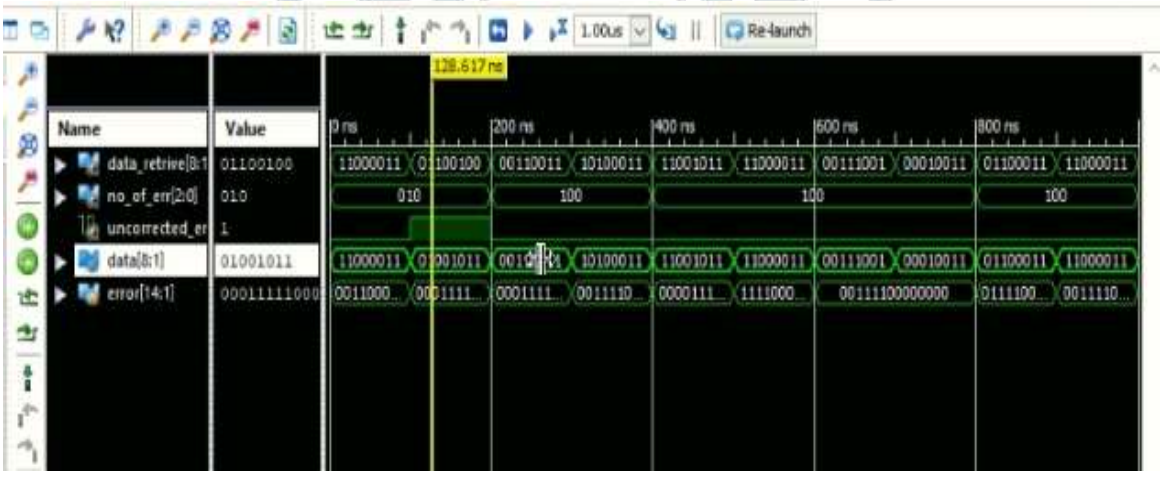

Figure 7: Error Transition from zero to one

Figure 7 is providing different input bits and out bits according the error and no error condition. Error transition is zero to one for a case, for others it's in zero.

Table 2: Simulation Parameters (Virtex 5)

Sr. No	Parameter	Value
1	Area	6.75%
2	Delay	2.538ns
3	Frequency	394MHz
4	Memory	4624652 kilobytes
5	Completion Time	20.00 Secs
6	Throughput	3.15 GHz
7	Power	450 µW
8	PDP	113.85 fJ

In table 2, simulation parameters are showing which is taken during the execution of verilog script.

Table 3: Comparison chart of proposed work with Base Work

Sr. No	Parameters	Previous Work	Proposed Work
1	No of Error	Single	Multi
2	Area	2826.59 μm^2	675 μm^2
3	Delay	434.4 ps	2.538 ns
4	Power	619.47 µW	450 µW
5	PDP	269.10 fJ	113.85 fJ

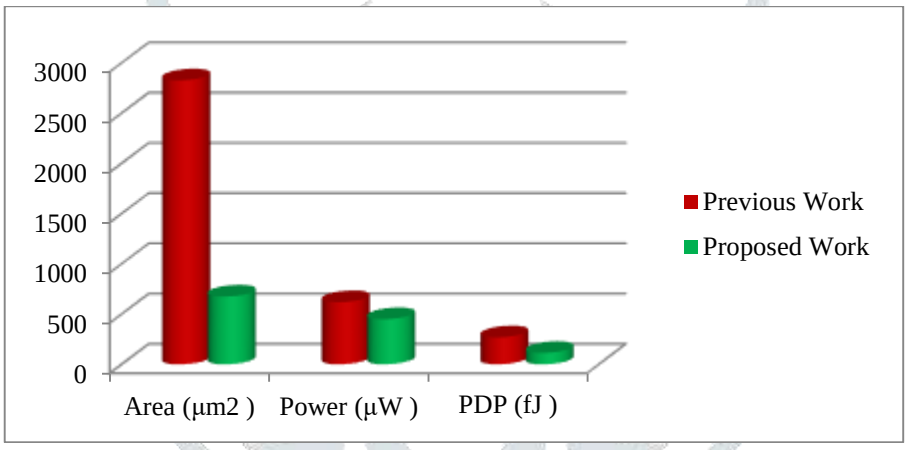

Figure 8: Comparison Graph

Figure 8 is shows the comparison graph of area, power and PDP of proposed and the previous work. It is clear from the comparison table and graphs. It can be say that the proposed work gives the significant better results than the previous work.

V. CONCLUSION

This work proposed an implementation of multi error correction and detection using verilog code on xilnx 14.7 software. Proposed MEC-MED circuit design for high speed and less complexity and reduced area logic circuit. The conventional SEC-SED logic is modified to extend error correction and detection and improve latency, area, power, throughput and frequency. The conditions for our simulations are: ISim simulator at vertex-v family. The comparison of proposed and previous work is done in terms of the calculated parameters. The proposed work utilized the 675 µm2 area while previous work utilized the 2826.59 µm2. The optimized delay is 2.538 ns while previous delay is 434.4 ps. The power consumption is 450 µW by proposed while 619.47 µW in previous. Simulated results shows that proposed MEC-MED VLSI architecture gives significant improved results than conventional SEC-SED.

REFERENCES

1. X. Peng, X. Liu, Y. Mei, J. Ren and H. Tang, "A Solution for Ultra-Low Bit-Error-Rate Interface of Superconductor-Semiconductor by Using an Error-Correction-Code Encoder," in IEEE Transactions on Applied Superconductivity, vol. 29, no. 5, pp. 1-4, Aug. 2019, Art no. 1301604, doi: 10.1109/TASC.2019.2900581.
2. S. Tripathi, J. Jana, J. Samanta and J. Bhaumik, "Fast and Power Efficient SEC-DED and SEC-DED-DAEC Codes in IoT based Wireless Sensor Networks," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 540-545, doi: 10.1109/TENCON.2019.8929354.

3. S. G. and R. N., "VLSI design of Parity check Code with Hamming Code for Error Detection and Correction," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 15-20, doi: 10.1109/ICCS45141.2019.9065790.
4. A. Zeh, M. Meier and V. Rieger, "Parity-Based Concurrent Error Detection Schemes for the ChaCha Stream Cipher," 2019 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Noordwijk, Netherlands, 2019, pp. 1-4, doi: 10.1109/DFT.2019.8875478.
5. C. Shao, H. Li, J. Fang and Q. Deng, "An Error Location and Correction Method for Memory Based on Data Similarity Analysis," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 10, pp. 2354-2364, Oct. 2019, doi: 10.1109/TVLSI.2019.2920755.
6. J. Gracia-Morán, L. J. Saiz-Adalid, D. Gil-Tomás and P. J. Gil-Vicente, "Improving Error Correction Codes for Multiple-Cell Upsets in Space Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 10, pp. 2132-2142, Oct. 2018, doi: 10.1109/TVLSI.2018.2837220.
7. Y. Zhang et al., "iRazor: Current-Based Error Detection and Correction Scheme for PVT Variation in 40-nm ARM Cortex-R4 Processor," in IEEE Journal of Solid-State Circuits, vol. 53, no. 2, pp. 619-631, Feb. 2018, doi: 10.1109/JSSC.2017.2749423.
8. R. G. Rizzo, V. Peluso, A. Calimera, J. Zhou and X. Liu, "Early bird sampling: A short-paths free error detection-correction strategy for data-driven VOS," 2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Abu Dhabi, United Arab Emirates, 2017, pp. 1-6, doi: 10.1109/VLSI-SoC.2017.8203472.
9. N. Sayed, F. Oboril, R. Bishnoi and M. B. Tahoori, "Leveraging Systematic Unidirectional Error-Detecting Codes for fast STT-MRAM cache," 2017 IEEE 35th VLSI Test Symposium (VTS), Las Vegas, NV, USA, 2017, pp. 1-6, doi: 10.1109/VTS.2017.7928937.
10. W. Jin, S. Kim, W. He, Z. Mao and M. Seok, "In Situ Error Detection Techniques in Ultralow Voltage Pipelines: Analysis and Optimizations," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 1032-1043, March 2017, doi: 10.1109/TVLSI.2016.2625598.
11. M. S. Reorda, L. Sterpone and A. Ullah, "An Error-Detection and Self-Repairing Method for Dynamically and Partially Reconfigurable Systems," in IEEE Transactions on Computers, vol. 66, no. 6, pp. 1022-1033, 1 June 2017, doi: 10.1109/TC.2016.2607749.
12. A. Abraham and M. Manuel, "Delay efficient error detection and correction of parallel IIR filters using VLSI algorithms," 2016 International Conference on Emerging Technological Trends (ICETT), Kollam, India, 2016, pp. 1-5, doi: 10.1109/ICETT.2016.7873655.
13. M. S. Sundary and V. Logisvary, "Multiple error detection and correction over GF(2m) using novel cross parity code," 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2016, pp. 1-6, doi: 10.1109/ISCO.2016.7726963.
14. R. Zhou and W. Qian, "A general sign bit error correction scheme for approximate adders," 2016 International Great Lakes Symposium on VLSI (GLSVLSI), Boston, MA, USA, 2016, pp. 221-226, doi: 10.1145/2902961.2903012.