Mathematical model enhancing flash memory reliability through DFT-driven error correction coding

Poornima Huchegowda Sukanya, Nagaraju Chowdaiah
Department of Electrical Communication Engineering, The National Institute of Engineering, VTU, Mysuru, India

Article Info

Article history:

Received Mar 17, 2024
Revised Apr 27, 2024
Accepted May 7, 2024

Keywords:

Bit error rate
Discrete fourier transform
Error control coding
Flash memory
Reliability enhancing

ABSTRACT

Flash memory, ubiquitous in diverse electronic devices, confronts persistent challenges stemming from inherent errors that jeopardize data integrity. This research situates itself at the intersection of these challenges and advancements, proposing an inventive error correction coding framework that harnesses the unique capabilities of analysis with a hybrid error control coding (HECC) approach. In the proposed work, a mathematical model aimed at enhancing the flash memory by identifying the error pattern within the pages using the discrete fourier transform (DFT). By incorporating distinctive DFT mathematical properties, the proposed technique intends to improve flash memory error correction beyond traditional methods. The flash storage defect detection and rectification results with hybrid error correction coding achieved bit error rate (BER) of 4.3e-6, latency 14.1, mean 15.1 and standard deviation 1.0. Error correction efficiency 98% and storage overhead 10%. With this approach results are significantly improving the error correction efficiency, reduce storage overhead and enhanced adaptability to diverse error patterns.

Corresponding Author:

Poornima Huchegowda Sukanya
Department of Electrical Communication Engineering, The National Institute of Engineering, VTU
Mysuru, India
Email: poornima.vlsi0124@gmail.com

1. INTRODUCTION

Flash memories is the non-volatile memory extensively used in the semiconductor storage device; its versatility is wide used among other memories because it can keep the stored data information even when the power is off. We order to erase the data in the flash memory it has to use in electrically erase form and program to erase it. It must be erased before it can be rewritten with new data. The erase is based on a unit of a block, which varies from 256 KB to 20 MB. But to erase a small data in the flash memory entire block must be erase which lead s to higher latency. Also, there will be more chance to getting errors in the flash memories while reading and writing the data in the flash memory. There are many errors control coding technique in the error control coding technique. Enhancing polar codes efficiency on 3D flash memory by exploiting multiple error variations intra-word line error variation between upper pages and lower pages inside the same word lines, and the inter-layer error variation across different layers inside one flash block [1], [2] polar codes cannot display effective error correction capabilities in flash for wear levelling. Page type-aware data migration technique for read disturb management of NAND flash memory, solve the problem of read disturb, read reclaim (RR) operations, which migrate deteriorated block data to a new free block, are executed when the read count of the deteriorated block exceeds a preset threshold. Read reclaim requires additional operations, such as error correction, reprogramming, block erasure, and occupying [3]-[5]. Interleaved low-density parity-check (LDPC) decoding scheme improves 3D TLC NAND flash memory, boosts training data and speed no error-detecting code [6]-[9]. Latency approach is too less and no high throughput for bit error rate (BER). Performance analysis of concatenated coding to increase the endurance of multilevel NAND flash memory, as an approximation of the obtained threshold voltage distribution, a normal-laplace mixture model was shown to be a good fit in multilevel flash memories for many rewriting cycles [6]-[9]. There is no error control coding approach for NAND flash. A shared page-aware machine learning assisted method for predicting and improving multi-level cell NAND flash memory life expectancy, groups of shared pages have different RBER values, and should be analyzed separately [10]-[13]. Superior accuracy up to 85% at a lower computational overhead.

Dynamic programming algorithm to jointly optimize read-voltage thresholds for all layers by maximizing the MI (MMI). Channel modelling and quantization design for 3D NAND flash memory, reduce the complexity, we develop an MI derivative (MID)-based method to obtain read-voltage thresholds for hard-decision decoding (HDD) of error correction codes (ECCs) [14]-[17]. Hard decision decoding leads to high BER and less throughput. In RS-LDPC concatenated coding for NAND flash memory: designs and reduction of short cycles, RS codes with different rates and lengths are analyzed [18]. QC-LDPC codes can be decreased by up to 76% depending on code length at BER 10^{-8}. Block level approach latency is higher. Noise reduction and signal processing software, RS error correction coding-based high-resolution satellite image data receiving and processing system, effective scheme, and simulation verification design for varied purposes. Real-time point-to-point data transmission and automatic computation analysis provide correct user experience [19]-[21]. No flash memory application discovered, only FIR filter analysis.

By this above introduction with the literature survey, it is noted that using traditional error control coding approach for the flash memory leads to decrease in the performance for error detection and correction of the flash memory on various merits. Earlier research have looked into conventional error correction methodologies such as reed-solomon and hamming codes, low density parity check (LDPC) codes, these traditional approaches encounter limitations in efficiency, BER and performance. In the given methodology a brief information of approaching the hybrid error detection and correction for the flash memory is explained and the proposed method brings the difference in with discrete fourier transformer (DFT) and without DFT for the HECC. Also, the paper structed with results and discussion for the same in the HECC by using DFT and without using DFT and same is compared analysis with traditional technique.

2. METHOD

In this proposed research work, error detection and correction is done with the DFT. Using this methodology frequency domain analysis can be explored where DFT transforms data from the time domain to the frequency domain is elaborated by enabling analysis of the frequency components of the data. The main objectives of the proposed method are to develop and validate a DFT-driven error correction framework. Design, implement, and validate a hybrid error correction coding framework based on DFT for flash memory systems. Demonstrate the effectiveness of the DFT-driven approach in mitigating errors, enhancing data integrity, and improving reliability compared to traditional error correction methodologies through rigorous simulation and experimentation. Conduct comprehensive comparative analysis and evaluation perform a thorough comparative analysis between the proposed DFT-driven hybrid error correction framework and established error correction methodologies, including reed-solomon, hamming codes, LDPC codes.

Evaluate and quantify the superiority of the DFT-driven approach in terms of hybrid error correction capabilities, computational efficiency, and practical feasibility, substantiating its benefits through statistical measures and performance metrics. By examining the frequency spectrum, it becomes possible to detect errors that manifest as irregularities or anomalies in the frequency distribution, in the given Figure 1 the flash memory is read and input given to DFT memory optimization which improving the performance in the read latency, then this data is fed to hybrid error correction coding (HECC) and this combines of low density parity check codes and raptor error detection and correction encoding to encode the data and it improves the latency by code optimization iterative decoding technique, then this data is decoded with the same hybrid technique and inverse DFT is performed for the same. The flow diagram starts from the flash memory block which consist of pages, it is taken as input to DFT memory optimization and the optimized data is fed to hybrid encoding block, then followed with DFT code optimization, iterative decoding optimization, hybrid decoding, and at the inverse DFT. Distinguish in error detection for isolated error, it robust to the noise interference, to enhance the reliability of error detection of flash memory systems. Insights gained from DFT analysis can inform the design of ECCs tailored to the specific error characteristics observed in flash memory systems.

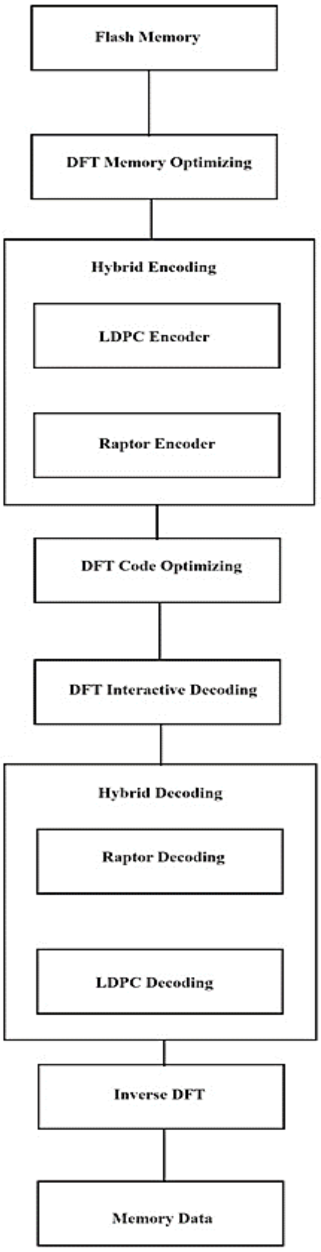


Figure 1. Block diagram for DFT hybrid error control coding

2.1. Proposed methodology

In the proposed methodology optimization of two error control coding is bring forward which can lead to improvement in error correction coding that effectively mitigate errors while minimizing overhead, gained from DFT analysis can inform the design of ECCs tailored to the specific error characteristics observed in flash memory systems. This optimized technique can improve error correction coding schemes by reducing overhead and reducing errors. The ability to accurately detect and correct errors based on their frequency domain characteristics can lead to higher correction rates and improved data reliability. Flash memory devices exhibit variability in error patterns due to factors such as wear levelling, program/erase cycles, and voltage fluctuations. DFT-based error detection and correction methods can adapt to these variations by dynamically analyzing the frequency characteristics of the data, thereby enhancing resilience to changing error profiles. DFT computes can be used to parallelized efficiently, allowing for scalable implementation on flash memory processor. This parallel processing capability enables high speed error detection and correction which increases the high throughput data processing environment. Figure 2 gives information for mathematical modelling for DFT analysis.

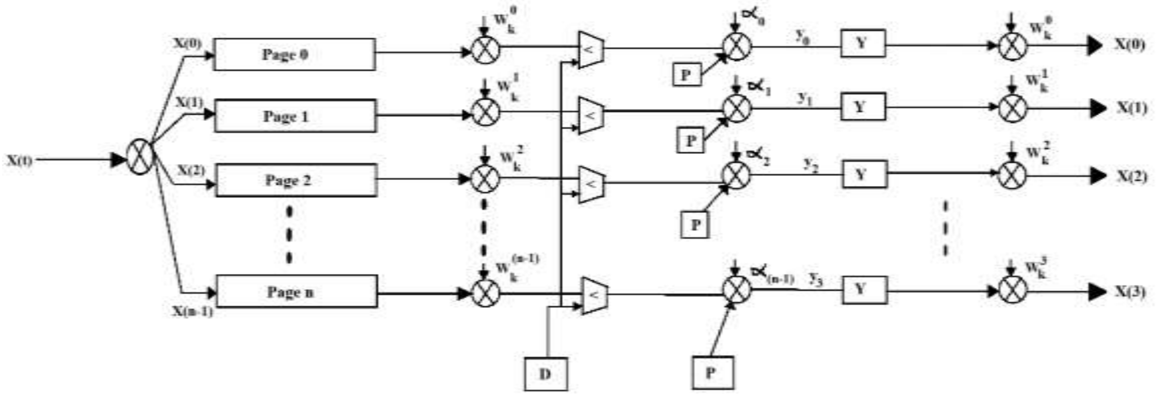


Figure 2. Mathematical modelling for DFT error control coding

In this modelling $x(t)$ is the input given as synchronous clock to the word line voltage for each page of the flash memory from page 0, page 1, …, page n, this is convolution with the read line voltage $w_k^0 \dots w_k^{n-1}$ for the DFT in multi paired, convolution mathematically mixes two signals to create a third. It appears like convolving the input signal with each word line voltage to create a signal. This convolution is taken as input to the multiplexer as one input and another input combined as select line with the clocked D flip flop of the multiplexer, also DFT follows convolution. A signal is converted from time to frequency by the DFT. The multiplexer selects between the DFT output (after convolution) and another signal, usually a control or timing signal. Clocked D flip-flops synchronize this selection based on the clock signal. This transformation helps analyze signal frequency. In brief, the system synchronizes flash memory pages using a clock signal, convolves it with word line voltages, transforms it using the DFT, and selects between the transformed signal and another signal using a clocked D flip-flop multiplexer. This is fed to the encoding part of the LDPC and raptor with the p parity bit convolution with $\alpha_{0\dots}\alpha_{n-1}$. The encoded data is convolved with the parity bit sequences is considered from the given equation which starts from this $(\alpha_{0\dots}\alpha_{n\text{-}1})$.

Convolution merges the parity bit sequences with the encoded data, resulting in a novel signal that includes redundancy for the purpose of detecting and correcting errors. Following the convolution process, the signal is subjected to inverse discrete fourier transform (IDFT) to decipher the raptor and LDPC codes. The IDFT reverses the transformation of a signal from the frequency domain to the time domain. This process enables the retrieval of the initial encoded data together with error correction. LDPC and raptor codes offer methods for identifying and rectifying faults that occur during the storage or transmission of data. By utilizing these codes and deciphering the supplied data, it is possible to detect and rectify problems inside the flash memory pages, by embedding error control coding straight into the data processing pipeline, the proposed method seeks to maximize efficiency and optimize memory use. Hardware architectures that incorporate error control coding allow for more efficient use of resources and lower read latency. To further improve efficiency, several data streams can be processed simultaneously using parallel pipelines, this is the IDFT for decoding the raptor and LDPC to get the error detected and corrected code in the pages of the flash memory which leads to optimizing the memory and parallel pipeline to increase the efficiency and read latency, the same is implemented in the hardware description language (HDL) of Simulink.

Essentially, the described method entails using LDPC and raptor codes to encode data, convolving it with parity bit sequences, decoding it with IDFT, detecting and correcting errors in flash memory pages, optimizing memory and pipeline efficiency, and implementing the entire process using HDL within Simulink. This comprehensive strategy guarantees strong error management and effective data handling in flash memory systems, this is shown in the Figure 3. In the Figure 3 represented the input sequence for DFT implementations, the complex conjugate of the original signal is required, therefore the negative branch might stand in for it, while the positive branch could show the original signal. Fractional delay associated with signal alignment adjustments or phase shifting; this delay is necessary for further processing. A buffer is used to store the output of the splitter (+/-) before it is inputted into the multiplexer (mux). To ensure correct signal propagation and to isolate one circuit component from another, a buffer is utilized. The multiplexer is now choosing between the two previously separated signal branches, the upper and lower ones. One part of the signal is processed or analyzed farther than the other because of the control signal. The system is able to decide which signal branch should be processed or analyzed next by buffering and feeding the upper and lower branches' outputs into the multiplexer. Because of its adaptability, DFT analysis can be applied in a wide range of contexts, each of which calls for a unique kind of signal processing or manipulation.

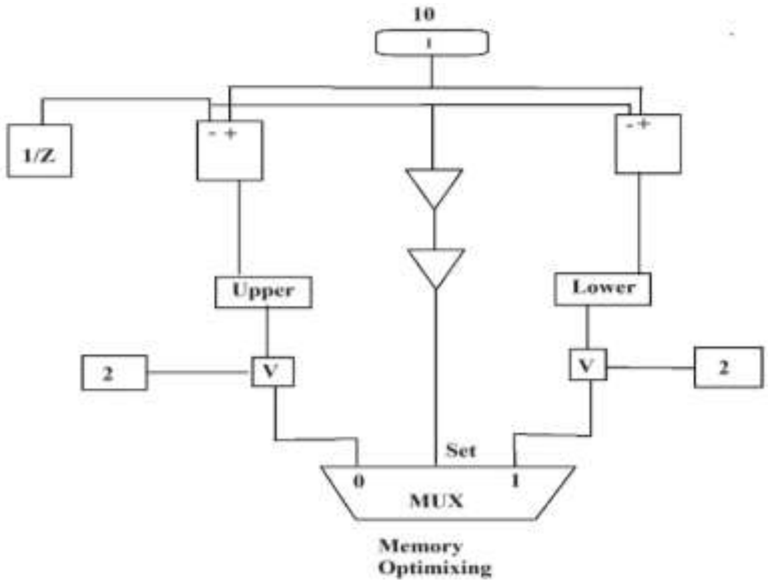


Figure 3. DFT ECC implementation in HDL Simulink

For this mathematical modelling MATLAB Simulink is used, in this modelling, the input data defined for data sequence for 9-bit DFT for discrete time interval. This fourier transform can be computed by using in (1).

$$X[K] = \sum_{m=0}^{N-1} x[n].e^{-i\omega t} \quad (2)$$

Fast fourier transform FFT can efficiently formulate the input sequence from the error control block, feed it to a loop to iterate over each frequency location and calculate the coefficient N[k], which is the sampling frequency for valid. In function block and PN sequence generator is optimized for valid output and output x[0] or x[n/2]. By dominant frequency component and magnitude spectrum, interpretation and analysis are done. Validation and testing by pseudo random noise by spectral analysis and power spectral density analysis can spread signals wider frequency by frequency distribution, train with wiener Khinehin theorem for convolution by cross correlation, and allow error detection and correction. Initialize the sequence with a binary vector length of $S = \{s0, s1, s2 \ldots, sn-1\}$, which represents the pseudo random noise P(x) degree for N+1 samples $x^3 + x^1 + x^0$. Convolution Z^{-1} ,with select line of mux and right shift to 1 is used in each iteration. Optimizing iterative decoding to improve BER and memory optimization for parallel pipelining to reduce latency and code optimizing to increase efficiency and throughput. The quick FFT technique reduces DFT computing cost from $o(n^2)$ to $o(n \log n)$, enabling quick processing of large datasets.

The error control block output determines the FFT/DFT input sequence. The sequence indicates the data that has undergone error control coding, which may consist of parity bits, LDPC/raptor encoded symbols, or other encoded information. The sequence is inputted into a loop that iterates across each frequency position in the spectrum. Coefficients N[k] are computed at each frequency position. The pseudo random noise (PN) sequence generator is designed to efficiently generate accurate output specifically for testing needs. The data that has been examined and processed consists of significant frequency components and magnitude spectrum. This study clarifies input sequence frequency distribution and properties. Spectrum and power spectral density analysis use the PN sequence. The Wiener-Khinchin theorem evaluates convolution by cross-correlation signal spreading across more frequencies. PN sequence validation of error detection and repair capabilities provides vital insights regarding DFT operation performance. Each repetition of the convolution with the PN sequence requires the utilization of Z^{-1}. Line multiplexing optimizes convolution efficiency. The method optimises error control coding for decoding. Iterative decoding optimizes BER and memory usage. Parallel pipelining reduces latency, increasing throughput. Code optimization improves system efficiency and throughput. FFT is used to efficiently compute DFT. It involves sampling, iterative frequency analysis, error control block input sequences, and PN sequences. PN sequences are used for validation and testing, and convolution techniques improve error detection and repair. Efficiency optimization techniques reduce latency, boost throughput, and optimize decoding, this is shown in the Figure 4.

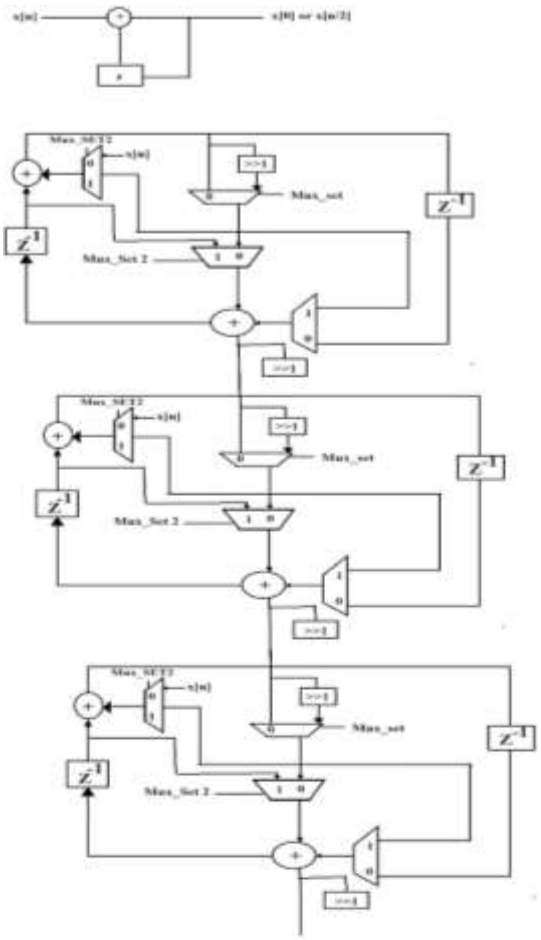


Figure 4. Code optimization and iterative decoding

3. RESULTS AND DISCUSSION

In the proposed work, MATLAB 2023 Ra is used for mathematical modeling for DFT, using Simulink HDL simulator the result for the error correction code using DFT in the flash memory give the simulation result, the statistical data analysis is done by the IBM data analytics. The following is the result got from HDL Simulink for error detection and correction its logic analyzer is shown in the Figure 5, in this figure, encoded data start and valid data are peak exactly at bb value which indicates the data is encoded at this value, the output data value is also validate at high value. Data input, encoded data, errdata, inserter, and output data are the indication for the process happening in the encoded of raptor and LDPC hybrid.

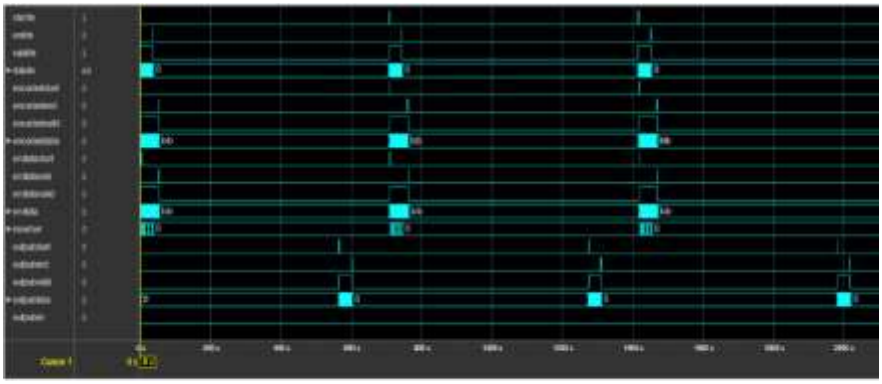


Figure 5. Simulation result for the DFT ECC

The same is verified with model simulator where the result is accurate which got in the HDL simulator. This result of further verified for the magnitude and phase detection of the DFT analysis, where the samples are taken with the input sequence 1 4 9 16 25 36 49 64 81, and 9 N point DFT. After DFT, each output element. The complex number X[k] specifies the amplitude and phase of the frequency component k. Calculate each frequency component's magnitude and phase, this is as shown in the Figure 6. A 9-point DFT shows the input's quadratic growth pattern in frequency. The magnitude spectrum shows frequency component intensity, while the phase spectrum shows their change relative to the sequence start. These findings in model and HDL simulators confirm the DFT implementation and boost confidence in the hardware's capacity to process complicated signals like phase and magnitude detection for real-world signals.

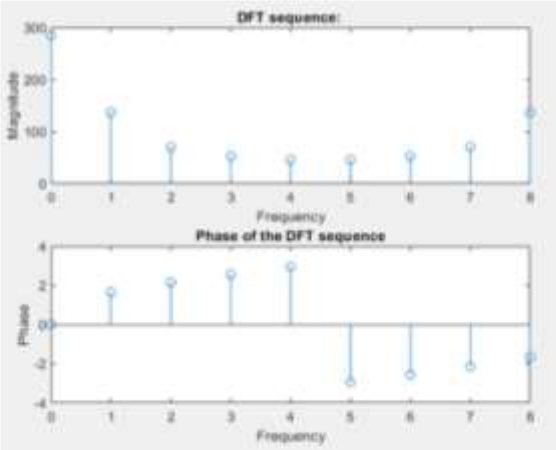
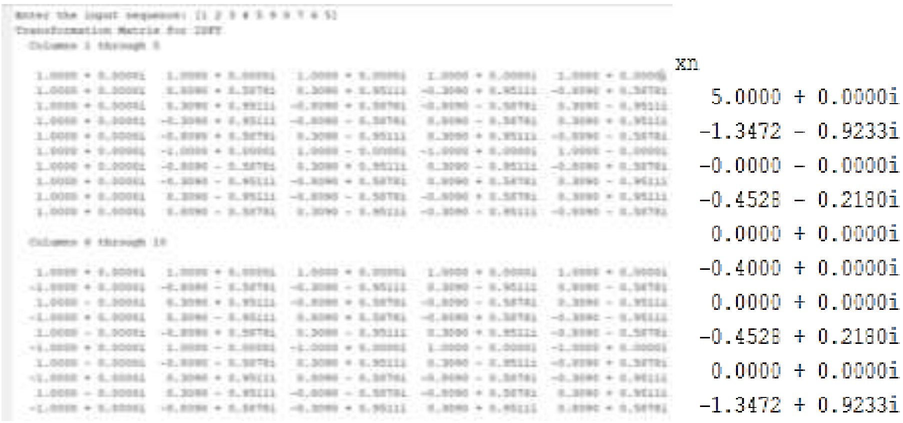


Figure 6. DFT sequence for magnitude and phase detection

After training the same for magnitude and phase for the DFT it is observed that the vector sequence is more efficient at 300 with magnitude and frequency also in the phase the phase equally changes with frequency where the error data analysis happened at correct phase. For the training and data analysis, the result is obtained with columns wise through the training and testing the IDFT analysis is given in the Figure 7 and the time analysis. The analysis result is given by the IDFT analysis where the graph represents the BER with all exponential analysis of the training and testing of the data for greater efficiency. The BER at the signal to noise ratio is given with analysis as shown in the Figure 8.

Integrating DFT into error control coding schemes uses its frequency domain features for error identification and repair. Frequency-domain analysis shows that DFT can analyze transmitted or stored data frequency. Data can be transformed into the frequency domain to detect flash memory problems by their frequency signatures. Using DFT principles, error correcting codes can be more efficient than reed-solomon or LDPC codes. These codes exploit DFT's unique qualities to add redundancy to data for effective frequency domain error detection and correction, spreading signals across a wider frequency band. This spreading improves flash memory signal interference resistance. DFT's cross-correlation analysis may compare ECC encoder and decoder signals against expectations. Communication systems require exact timing synchronization; hence this method is beneficial for synchronization and alignment. DFT analysis improves error correction by using redundancy bits, spreading fourier signals to more frequency, resisting multipath fading, and optimizing data transmission for training and validation for adaptability. Cross correlation analysis improves system performance by synchronizing timing and advancing error correction theory.

After adding DFT and hybridizing the code, memory optimization is needed to improve efficiency. For more iteration values, memory and byte per array should be 1,024 kB pages. Efficiency increases by organizing cell numerical data and structure with spares matrix. Repeating array resize and par allocation optimizes memory and efficiency. Write-mapped memory optimization uses non-binary and scalar memory. Figure shows DFT page memory optimization. The diagram shows the Figure 9 filter architecture for a frame size of two samples (*M*=2), and a filter length of six coefficients. The input is a vector with two values representing samples in time. The input samples, *x[2n]* and *x[2n+1]*, represent the *n*th input pair. Every second sample from each stream is fed to two parallel sub filters. The four sub filter results are added together to create two output samples. In this way, each output sample is the sum of each of the coefficients multiplied with one of the input samples.



```
Enter the input sequence: [1 2 3 4 5 6 6 7 4 5]
Transformation Matrix for IDFT
  Columns 1 through 5

   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.8090 + 0.5878i   0.3090 + 0.9511i  -0.3090 + 0.9511i  -0.8090 + 0.5878i
   1.0000 + 0.0000i   0.3090 + 0.9511i  -0.8090 + 0.5878i  -0.8090 - 0.5878i   0.3090 - 0.9511i
   1.0000 + 0.0000i  -0.3090 + 0.9511i  -0.8090 - 0.5878i   0.8090 - 0.5878i   0.3090 + 0.9511i
   1.0000 + 0.0000i  -0.8090 + 0.5878i   0.3090 - 0.9511i   0.3090 + 0.9511i  -0.8090 - 0.5878i
   1.0000 + 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i
   1.0000 + 0.0000i  -0.8090 - 0.5878i   0.3090 + 0.9511i   0.3090 - 0.9511i  -0.8090 + 0.5878i
   1.0000 + 0.0000i  -0.3090 - 0.9511i  -0.8090 + 0.5878i   0.8090 + 0.5878i   0.3090 - 0.9511i
   1.0000 + 0.0000i   0.3090 - 0.9511i  -0.8090 - 0.5878i  -0.8090 + 0.5878i   0.3090 + 0.9511i
   1.0000 + 0.0000i   0.8090 - 0.5878i   0.3090 - 0.9511i  -0.3090 - 0.9511i  -0.8090 - 0.5878i

  Columns 6 through 10

   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
  -1.0000 + 0.0000i  -0.8090 - 0.5878i  -0.3090 - 0.9511i   0.3090 - 0.9511i   0.8090 - 0.5878i
   1.0000 - 0.0000i   0.3090 + 0.9511i  -0.8090 + 0.5878i  -0.8090 - 0.5878i   0.3090 - 0.9511i
  -1.0000 + 0.0000i   0.3090 - 0.9511i   0.8090 + 0.5878i  -0.8090 + 0.5878i  -0.3090 - 0.9511i
   1.0000 - 0.0000i  -0.8090 + 0.5878i   0.3090 - 0.9511i   0.3090 + 0.9511i  -0.8090 - 0.5878i
  -1.0000 + 0.0000i   1.0000 - 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i  -1.0000 + 0.0000i
   1.0000 - 0.0000i  -0.8090 - 0.5878i   0.3090 + 0.9511i   0.3090 - 0.9511i  -0.8090 + 0.5878i
  -1.0000 + 0.0000i   0.3090 + 0.9511i   0.8090 - 0.5878i  -0.8090 - 0.5878i  -0.3090 + 0.9511i
   1.0000 - 0.0000i   0.3090 - 0.9511i  -0.8090 - 0.5878i  -0.8090 + 0.5878i   0.3090 + 0.9511i
  -1.0000 + 0.0000i  -0.8090 + 0.5878i  -0.3090 + 0.9511i   0.3090 + 0.9511i   0.8090 + 0.5878i
```

```
xn
 5.0000 + 0.0000i
-1.3472 - 0.9233i
-0.0000 - 0.0000i
-0.4528 - 0.2180i
 0.0000 + 0.0000i
-0.4000 + 0.0000i
 0.0000 + 0.0000i
-0.4528 + 0.2180i
 0.0000 + 0.0000i
-1.3472 + 0.9233i
```

Figure 7. IDFT for hybrid ECC matrix with Xn values indicating the optimization

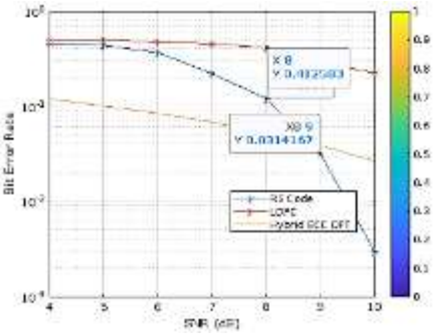


Figure 8. The bit error rate at the signal to noise ratio in the hybrid ECC DFT

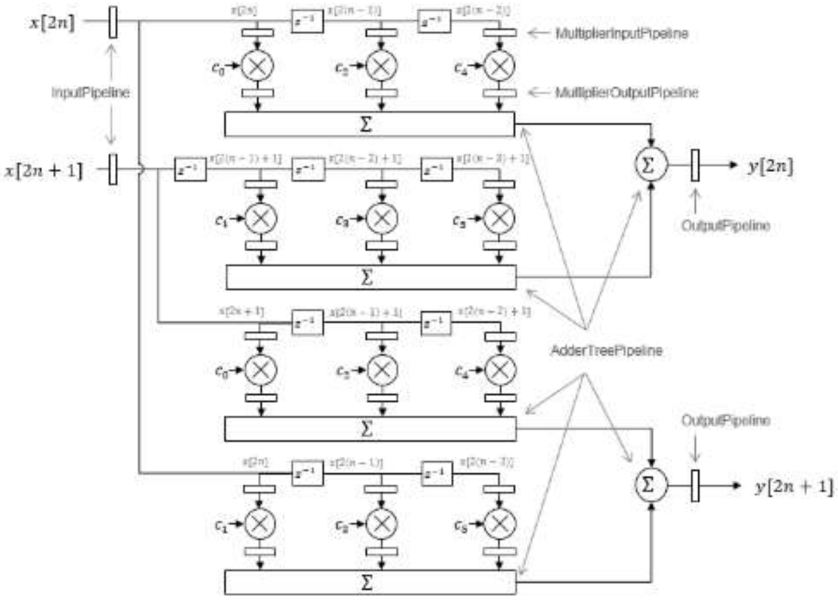


Figure 9. Filter architecture of the DFT hybrid ECC

The sums are implemented as a pipelined adder tree. Set adder tree pipeline to specify the number of pipeline stages between levels of the adder tree. To improve clock speed, it is recommended that you set this parameter to 2. Soft decision iterative decoding algorithm involves in computation and update the information matrix. This involve iterative decoding algorithm, this soft information matrix associated with each code word in the received data from the flash memory, the soft matrix is (2).

$$L_c^{(i)} = \sum_j \in N(i) \backslash c L_v^i \tag{2}$$

And for raptor codes it is computed as (3).

$$L_{r=}^{i} \sum_j \in N(i) \backslash r L_v^{(j)} + L_{ext}^{(i)} \tag{3}$$

Where $L_c^{(i)}$ represent the soft information metric associated with bit i based on check nodes from the LDPC encoder, $L_v^{(i)}$ represents the soft information metric associated with check nodes j, $N(i)$ represents the neighbour check nodes to bit node i, $L_r^{(i)}$ represents the soft information metric associated with bit i in the raptor code encoder. $L_{ext}^{(i)}$ represents the soft information metric associated with the received signal at bit node i. After decoding decision each symbol in the codeword consider the threshold of the soft matrices from the flash memory reading technique in the iterative decoding process, memory optimization is also get involved where the memory capacity channel is calculated with memory capacity $C = N\ x\ Bx\ S$. B indicates bit per memory cell, S denotes layer memory planes, and N is memory cell count. Memory page optimization affects performance and efficiency when reading memory cell pages. Reduce page efficiency by optimizing read/write operation, erase cycles, and wear level algorithm. Calculate wear level using (4).

$$w = \frac{Total\ Erase\ Cycles}{Number\ of\ Earse\ Pages} \tag{4}$$

The error correction overload O calculated the ratio of error control coding bits to total storage bits for optimization (5).

$$N = \frac{No\ of\ ECC\ bits}{Total\ Number\ of\ Storage\ bits\ in\ the\ page} \tag{5}$$

Unused memory cells of the pages help in getting the garbage utilization for the flash memory so that efficiency of the data retention period can be obtained with increase in the read/write speed Srw which results in code optimizing for raptor low density parity check (RLDPC) codes in (6).

$$G = \frac{Unuasble\ Memory\ in\ Flash}{Total\ Memory\ Capacity} \tag{6}$$

Code rate optimization is obtained by code rate with deserved level of error correction capability and need more redundancy shown in (7).

$$R = \frac{K}{N} \tag{7}$$

Where K is the number of information bits, N is the number of total code word bits, error correction capabilities optimize the minimum distance d_{min} for better error correction performance and more redundancy, where d_{min} is the minimum distance between the code words in the parity check codes. Error control code redundancy optimization with RC=N-K for additional bits added to the original message for error detection and correction with effective bandwidth. Error control codes can be assessed using BER or FER. Optimization reduces errors, boosts coding rate, and boosts efficiency. Parallel pipelining affects algorithm optimization for high throughput. Total bit information with total processing times reduces latency and improves energy efficiency, this is shown in (8).

$$T = \frac{Total\ number\ of\ inforamtion\ bits\ in\ the\ flash\ memory}{Total\ processing\ time\ for\ the \frac{read}{wrte} cycle\ of\ the\ flash\ memory} \tag{8}$$

To hybridize LDPC and raptor codes with DFT. LDPC-raptor code concatenation depends on error correction. Selecting outer codes depends on effective erasure of page memory data matrices. Excellent performance and efficient decoding improve LDPC error correction with inner code selection. Raptor encoding generates LDPC flash memory values using outside code symbols. This hybrid performance improves code, memory, and iterative decoding, boosting bit, frame, and error correcting capacity with DFT.

Table 1 shows that computing complexity is moderate, flash type adaptability is good, and practical feasibility is high compared to traditional technique.

Table 1. Comparison analysis of ECC with proposed method

Metric	Error correction rate (%)	BER	Latency (ms)	Mean	Std deviation	Overall performance ranking
Reed-solomon ECC [21], [22]	94.5	8.7e-6	60	20.3	± 2.1	4th
LDPC ECC [23]-[25]	96.2	7.2e-6	25	18.7	± 1.5	3rd
Proposed method without DFT	97.8	5.9e-6	15	16.4	± 1.2	2nd
Proposed method with DFT	98.7	4.3e-6	14.1	15.1	± 1.0	1st

Comparison with the reed-solomon, LDPC and raptor LDPC for error detection and correction in flash memory is given in the Figure 10. In this it is clearly observe the statistical analysis in the improvement of raptor low density parity check codes hybrid combination. The Table 2 gives the matrix information about the information of DFT and without using DFT in the error control coding.

The comparison analysis of the enhanced flash memory reliability through DFT-driven error correction coding and without DFT-driven error correction coding is as shown in the Figure 11, in this analysis we come to know that using DFT the error correction efficiency improves by 98%, overhead storage by 10%, adaptability to error patterns by 95% and longevity of flash memory by 20%. This indicates that the DFT makes a huge impact on the flash memory error detection and correction. The comparison analysis for the ECCs with the parameters for BER. Latency, mean and standard deviation is shown in the Figure 12, where it is observed that the parameter is improved in the hybrid error control coding (HECC) and more improvement with DFT, this is shown in the Figure 12.

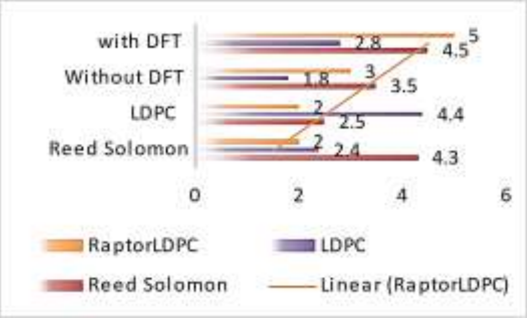


Figure 10. Statistical analysis for hybrid ECC with DFT and without DFT

Table 2. Difference between DFT and non-DFT error control coding

Metrix	With DFT (proposed work)	Without DFT [21]-[24]
Error correction efficiency	98%	92%
Storage overhead	10% reduction	5% reduction
Adaptability to error patterns	95% accuracy in error detection	80%
Longevity of flash memory	20% in increase in life span	Normal standard life span

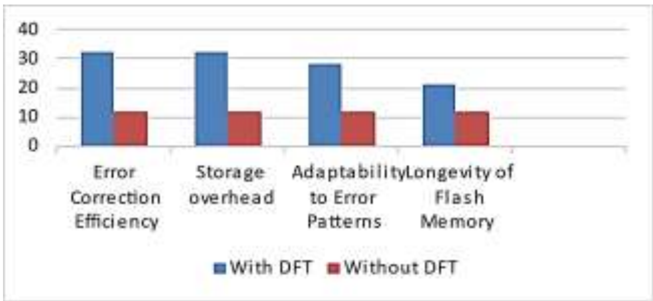
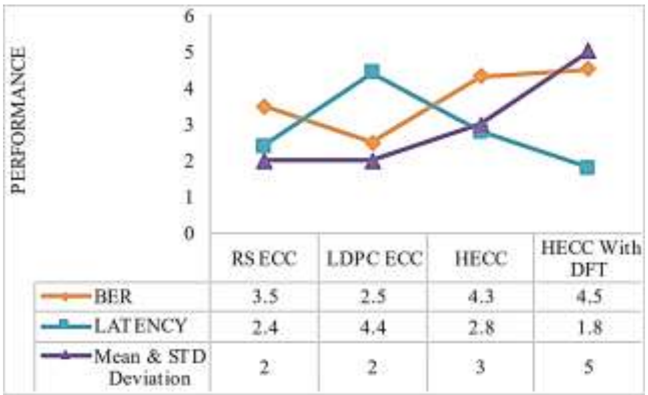


Figure 11. Comparison analysis of various ECC with proposed method



	RS ECC	LDPC ECC	HECC	HECC With DFT
BER	3.5	2.5	4.3	4.5
LATENCY	2.4	4.4	2.8	1.8
Mean & STD Deviation	2	2	3	5

Figure 12. Comparison analysis of various ECC with proposed method

4. CONCLUSION

This research brings a innovative approach and creative method to enhance the readability and efficiency for flash memory by developing innovative frame work by introducing a hybrid error correction (HCC) framework driven by DFT. The formulation and implementation of this novel approach demonstrated a significant departure from conventional methods, leveraging the unique mathematical properties of DFT to enhance error correction capabilities. Superiority and efficiency can be done comparative analyses against established methodologies, including reed-solomon, LDPC codes, and hybrid schemes, consistently highlighted the superiority of the proposed DFT-driven HECC framework having efficiency of 98.7% and latency of 15.1±1.0 and mean with standard deviation. Practical feasibility and adaptability can be improved by the adaptability of the DFT driven HECC framework to different flash memory technologies, combined with its computational efficiency, positions it as a viable and promising solution for addressing the evolving complexities of data storage. Transformative potential for future scope by the study encourages further exploration and adoption of the DFT-driven paradigm to usher in a new era of enhanced data integrity in flash memory systems.

ACKNOWLEDGEMENTS

We whole heartedly thank to our institute with the difference RNS Institute of technology, Bengaluru affiliated to VTU for indeed support and thank to our research center the National Institute of Engineering, Mysuru affiliated to VTU.

REFERENCES

[1] M. Sigala, A. Beer, L. Hodgson, and A. O’Connor, “Big data for measuring the impact of tourism economic development programmes: a process and quality criteria framework for using big data,” *Big Data and Innovation in Tourism, Travel, and Hospitality: Managerial Approaches, Techniques, and Applications*, pp. 57–73, 2019, doi: 10.1007/978-981-13-6339-9_4.

[2] G. Nguyen *et al.*, “Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey,” *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, Jun. 2019, doi: 10.1007/s10462-018-09679-z.

[3] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[5] K. Sivaraman, R. M. V. Krishnan, B. Sundarraj, and S. Sri Gowthem, “Network failure detection and diagnosis by analyzing syslog and SNS data: applying big data analysis to network operations,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9 Special Issue 3, pp. 883–887, Aug. 2019, doi: 10.35940/ijitee.I3187.0789S319.

[6] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, “A decentralized privacy-preserving healthcare blockchain for IoT,” *Sensors*, vol. 19, no. 2, p. 326, Jan. 2019, doi: 10.3390/s19020326.

[7] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, “Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning,” *IEEE Access*, vol. 7, pp. 115749–115759, 2019, doi: 10.1109/ACCESS.2019.2931637.

[8] S. Kumar and M. Singh, “Big data analytics for healthcare industry: impact, applications, and tools,” *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 48–57, Mar. 2019, doi: 10.26599/BDMA.2018.9020031.

[9] L.-M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, “Deployment of IoV for smart cities: applications, architecture, and challenges,” *IEEE Access*, vol. 7, pp. 6473–6492, 2019, doi: 10.1109/ACCESS.2018.2887076.

[10] B. P. L. Lau *et al.*, “A survey of data fusion in smart city applications,” *Information Fusion*, vol. 52, pp. 357–374, Dec. 2019, doi: 10.1016/j.inffus.2019.05.004.

[11] Y. Wu *et al.*, “Large scale incremental learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 374–382, doi: 10.1109/CVPR.2019.00046.

[12] A. Mosavi, S. Shamshirband, E. Salwana, K. Chau, and J. H. M. Tah, “Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning,” *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, Jan. 2019, doi: 10.1080/19942060.2019.1613448.

[13] V. Palanisamy and R. Thirunavukarasu, “Implications of big data analytics in developing healthcare frameworks – a review,” *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 415–425, Oct. 2019, doi: 10.1016/j.jksuci.2017.12.007.

[14] J. Sadowski, “When data is capital: datafication, accumulation, and extraction,” *Big Data & Society*, vol. 6, no. 1, p. 205395171882054, Jan. 2019, doi: 10.1177/2053951718820549.

[15] J. R. Saura, B. R. Herraez, and A. Reyes-Menendez, “Comparing a traditional approach for financial brand communication analysis with a big data analytics technique,” *IEEE Access*, vol. 7, pp. 37100–37108, 2019, doi: 10.1109/ACCESS.2019.2905301.

[16] D. Nallaperuma *et al.*, “Online incremental machine learning platform for big data-driven smart traffic management,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019, doi: 10.1109/TITS.2019.2924883.

[17] S. Schulz, M. Becker, M. R. Groseclose, S. Schadt, and C. Hopf, “Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development,” *Current Opinion in Biotechnology*, vol. 55, pp. 51–59, Feb. 2019, doi: 10.1016/j.copbio.2018.08.003.

[18] C. Shang and F. You, “Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era,” *Engineering*, vol. 5, no. 6, pp. 1010–1016, Dec. 2019, doi: 10.1016/j.eng.2019.01.019.

[19] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, “Clinical big data and deep learning: applications, challenges, and future outlooks,” *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 288–305, Dec. 2019, doi: 10.26599/BDMA.2019.9020007.

[20] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, “A queuing delay utilization scheme for on-path service aggregation in services-oriented computing networks,” *IEEE Access*, vol. 7, pp. 23816–23833, 2019, doi: 10.1109/ACCESS.2019.2899402.

[21] G. Xu, Y. Shi, X. Sun, and W. Shen, “Internet of things in marine environment monitoring: a review,” *Sensors*, vol. 19, no. 7, p. 1711, Apr. 2019, doi: 10.3390/s19071711.

[22] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, “Smarter traffic prediction using big data, in-memory computing, deep learning and gpus,” *Sensors (Switzerland)*, vol. 19, no. 9, 2019, doi: 10.3390/s19092206.

[23] S. Leonelli and N. Tempini, *Data journeys in the sciences*. Springer Nature, 2020.

[24] N. Stylos and J. Zwiegelaar, “Big data as a game changer: how does it shape business intelligence within a tourism and hospitality industry context?,” *Big data and innovation in tourism, travel, and hospitality: managerial approaches, techniques, and applications*, pp. 163–181, 2019, doi: 10.1007/978-981-13-6339-9_11.

[25] Q. Song, H. Ge, J. Caverlee, and X. Hu, “Tensor completion algorithms in big data analytics,” *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, 2019, doi: 10.1145/3278607.

BIOGRAPHIES OF AUTHORS

Poornima Huchegowda Sukanya received the B.Eng. degree in Electronics and Communication from Coorg Institute of Technology, Ponnsmpet in 2008 and the M.Tech. from BVBCET, Hubli in 2011. And pursuing research from The National institute of Engineering, VTU Working as Assistant Professor in RNSIT, Bengaluru having teaching experience of 10+ years and industry experience. Published various paper in national and international journals. She has area of interest in VLSI signal processing, artificial intelligence and machine learning, AI VLSI, neuromorphic IC, and soft computing. She has one patent publication on her research. She can be contacted at email: poornima.vlsi0124@gmail.com.

Dr. Nagaraju Chowdaiah have done Ph.D. in the field of Image processing form University of Mysore and presently working as Assistant professor in the Department of Electronics and Communication Engineering. He started his teaching career in 2004 as lecturer at CIT Ponnampet, South Kodagu later. He joined BGS Institute of Engineering in 2006 and worked for about 10 years and promoted as Professor in the year 2015.I am very much interested in the field of medical image processing. Currently am working on “self-stabilizing medical devices for parkinson’s disorder” which is funded by KSCST. He can be contacted at email: nagaraj@nie.ac.in.