Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 3, June 2026, pp. 2104~2115
ISSN: 2302-9285, DOI: 10.11591/eei.v15i3.10395

Faults detection in SOC using TCRMCN based MBIST with optimization based BISR

Madhava Rao Jillella, Prashanth Narayanappa Ananda
Department of Electrical and Electronics Engineering, BMS Institute of Technology and Management, Visvesvaraya Technological University, Belgaum, India

Article Info

Article history:

Received Apr 3, 2025
Revised Mar 26, 2026
Accepted Apr 19, 2026

Keywords:

Built-in-self-repair
Memory built-in self-test
Multi-objective fitness dependent optimization algorithm
System on chip
Temporal channel reconfiguration multi-graph convolution networks

ABSTRACT

Memory takes up much of the chip area in modern system on chip (SOC). It is more challenging to repair these memories with a traditional external equipment test method. In this paper, SOC devices using temporal channel reconfiguration multi-graph convolution networks (TCRMCN), based memory built-in self-test (MBIST), with optimization based built-in-self-repair (BISR) (TCRMCN-MBIST-BISR-SOC) is proposed. Further the results of the test are analyzed by TCRMCN, which detects different types of faults. Once faults are detected, the data is passed to BISR, where the faulty memory cells are replaced using redundant memory cells and optimized by multi-objective fitness dependent optimization algorithm (MOFDOA). The proposed method demonstrates significant improvements in delay, power consumption, and access time, outperforming existing approaches like adaptive dynamic k-nearest neighbor (ADKNN) fostered BIST and Namib beetle optimization approach (NBOA) espoused BISR for SOC-based devices (ADKNNF-BIST-NBOA-BISR), deep q-learning with bit-swapping-based linear feedback shift register fostered BIST and BISR for static random access memory (SRAM) (DQL-BSL-BIST-BISR), and design of a fast and energy-efficient MBIST architecture using Verilog (DF-EC-MBIST), the proposed method achieves 9.28%, 8.78%, and 9.29% higher accuracy while reducing delay by 9.45%, 5.36%, and 8.28%, respectively.

Corresponding Author:

Prashanth Narayanappa Ananda
Department of Electrical and Electronics Engineering, BMS Institute of Technology and Management
Visvesvaraya Technological University
Belgaum, India
Email: prashanthna@bmsit.in

1. INTRODUCTION

System on chips (SOCs) initially require memory to save a lot of data [1]. The memories, which were created utilizing complementary metal-oxide-semiconductor technology (CMOS), span a huge portion of the system-on-chips design. The lower size of the system-on-chips indicates that the larger outcome requires the memories. Each column as well as row repair is components of the memory repair method. There are two stages to memory healing mechanisms [2]. Through repair memory testing, the first phase confirms the failure identified by the memory built-in self-test (MBIST) controller. To fix the memory, the second phase obtains the repair signature. The repair signature is then applied using the scan chain of the repair register. Fuse box read-write test access port is regulated. The scan chains between the memory and the fuse are recorded in a particular repair log. The repair data is scanned while the greater voltage nut is being used. Reinstatement data is import and debugged in repair logs after an on-chip reset. Additionally,

redundancy holds each memory together [3]. Finally, the memory accuracy is tested by running the built-in self-test on renovated memory. Contrary to popular belief, memory is not made up of flip-flops and logic gates. To test memories in this situation, a variety of failure algorithms and test models are required. Any memory flaws have an impact on the overall performance of system-on-chips. Memory contains spare rows as well as columns. Damaged cells are replaced with spare columns or rows using the built-in self-repair logic. Moreover line/column repair or both are included in repair logic. To identify flaws in healthy memory cells, MBIST and repair tools quickly evaluate memory. The following fault models are suitable for memory testing: neighborhood design sensitive fault (NBSF), coupling, transition, stuck-at-fault, and address decoder faults [4].

The problem involves that SOCs memory flaws can have a substantial impact on overall performance, and it is still difficult to identify and fix these problems while controlling access time, power consumption, and area. Even with the use of repair mechanisms such as MBIST controllers and built-in-self-repair (BISR), effective memory testing and repair are still challenging problems. These drawbacks in the existing technique inspire us to perform this research study.

The novelty of the proposed temporal channel reconfiguration multi-graph convolution networks (TCRMCN)-MBIST-BISR-SOC approach lies in its integration of fault detection using TCRMCN with optimization-based memory repair through the multi-objective fitness dependent optimization algorithm (MOFDOA). This method improves fault classification accuracy and efficiently replaces faulty memory cells with redundant ones, enhancing the reliability, and performance of SOC devices under real-world conditions [5].

The main outlines of this research work are deliberated based:

- In this manuscript, SOC devices using TCRMCN based MBIST with optimization based BISR (TCRMCN-MBIST-BISR-SOC) is proposed.
- The collected data is processed through MBIST to detect defects in the memory cells by utilizing specific test patterns.
- The proposed TCRMCN approach enhances fault diagnosis by leveraging multi-graph learning to capture spatial-temporal dependencies in fault patterns, ensuring precise fault localization and classification. Furthermore, the method improves robustness by minimizing misclassification and enhancing adaptability to different memory in an array.
- After faults are detected, the data is sent to the BISR system, where defective memory cells are substituted with redundant ones and optimized using the MOFDOA.
- The proposed TCRMCN-MBIST-BISR-SOC method is evaluated with the existing methods ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively [6].

From the literature survey reviews, several BIST and BISR algorithms have been proposed, each offering advantages as well as disadvantages. The ADKNNF-BIST-NBOA-BISR, MBIST algorithm demonstrates high accuracy with low delay, making it suitable for time-sensitive applications. The DQL-BSL-BIST-BISR algorithm also offers high accuracy; however, it suffers from high power consumption, which could limit its use in power-constrained environments. In contrast, the DF-EC-MBIST algorithm shows low access time and reduced latency and power usage, though it comes with a trade-off of high delay. Another method, based on BISR for high bandwidth memory (HBM) with maximum fault collection and fast analysis, is noted for its low power consumption, but at the cost of reduced accuracy. As a result, a research gap is identified where the current algorithms can be further improved to minimize power consumption, decrease delay time, boost accuracy, and shorten access time [7].

The rest of this paper is arranged as follows. Section 2 covered proposed methodology. Section 3 presented the results and discussions. Section 4 describes the conclusion and future study directions.

2. METHOD

This section introduces the TCRMCN-MBIST-BISR-SOC methodology for fault detection and classification in static random access memory (SRAM) memory arrays. The process starts with acquiring data from the SRAM memory array, followed by MBIST-based fault detection, where targeted test patterns are applied to identify defective memory cells [8].

The detected faults are then processed using TCRMCN, which efficiently classifies different faults, enhances diagnostic accuracy and ensures reliable fault classification for improved SRAM performance and fault tolerance. Then the data is given to BISR, where faulty memory cells are replaced using redundant cells and optimized by an advanced optimization algorithm to enhance fault correction and system reliability. The block diagram of the proposed TCRMCN-MBIST-BISR-SOC methodology is shown in Figure 1.

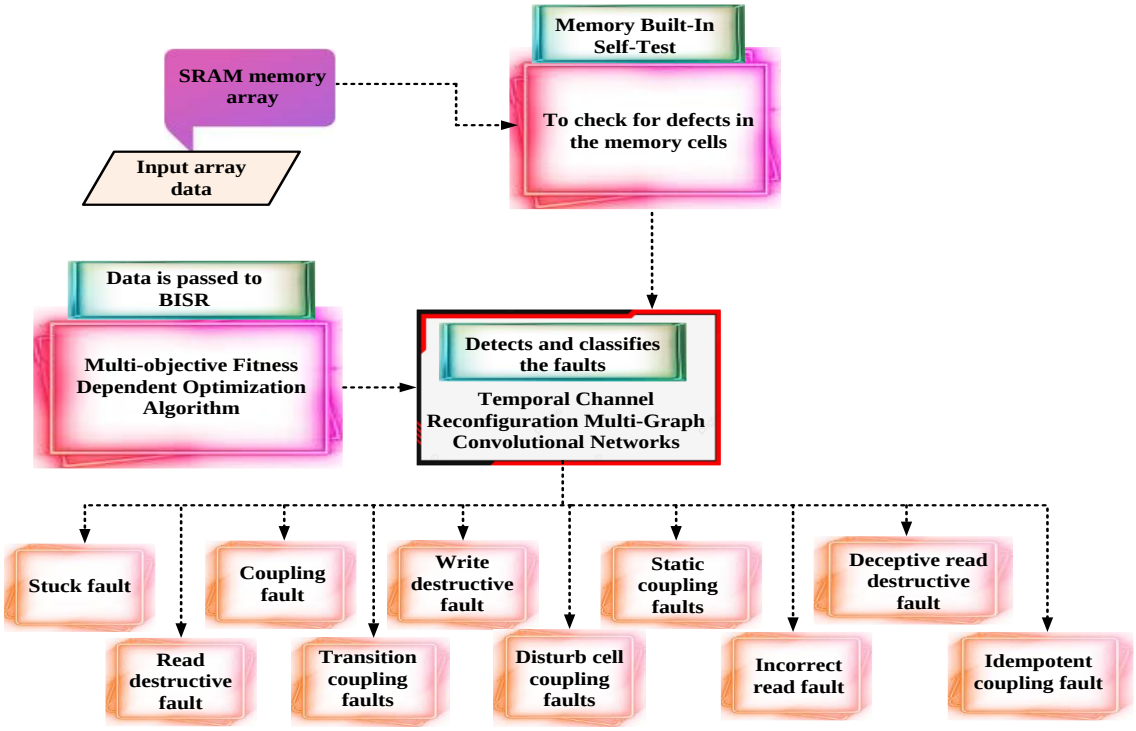


Figure 1. Schematic diagram of the proposed TCRMCN-MBIST-BISR-SOC system

The proposed methodology used to test the single port SRAM have 16×8 bit word length. TCRMCN-MBIST-BISR-SOC methodology is used to test and identify the various faults to analyze accuracy, delay, access time, and power consumption [9]-[11].

2.1. Data acquisition

The input data is collected from the SRAM memory array, where each memory cell is analyzed to identify various fault types. Stuck fault occurs when a memory cell remains locked at '0' or '1', while coupling fault arises due to interactions between adjacent cells, leading to unintended transitions. Read destructive fault and write destructive fault result in unintended flips during read and write operations respectively. Transition coupling fault is a dual-cell fault where a victim cell transitions due to aggressor writes, whereas static coupling fault occurs when an aggressor word forces a victim word into an incorrect state. Disturb cell coupling fault is a subset of coupling faults where a read or write operation disrupts an adjacent cell. Incorrect read fault leads to incorrect memory values during read operations, while deceptive read destructive fault initially returns correct values but inverts after a read operation. Idempotent coupling fault forces a victim word into an unintended transition due to an aggressor cell's influence.

2.2. Memory built-in self-test for defects in memory cells

In this section MBIST is discussed for defects in the memory cells using specific test patterns. MBIST enhances functionality by detecting faults in the memory interface ensuring reliable operation. It continuously monitors and verifies data integrity during read and writes operations identifying differences caused by defects. By utilizing built-in test mechanisms, MBIST reduces the need for external testing equipment, making fault detection more efficient and in maintaining system stability by preventing data corruption and enhancing memory reliability. Multiple input signature register (MISR) enhances error detection by compressing test responses and identifying faulty memory locations. The proposed MBIST engine for defects in memory cell diagram is presented in Figure 2.

Figure 2 illustrates the MBIST engine combines a defect detection unit with a multi-algorithm finite state machine (FSM) controller. The controller is implemented through sub-level modularity, enabled by states that interconnect dynamically based on the selected algorithm. These states are categorized into directional operations, where memory addresses progress or downward and simple operations that perform write one, read one, write zero, and read zero at the memory level. The controller includes an idle state and a stop state to regulate execution flow efficiently. This structured approach improves fault detection accuracy and ensures optimized memory testing for various architectures.

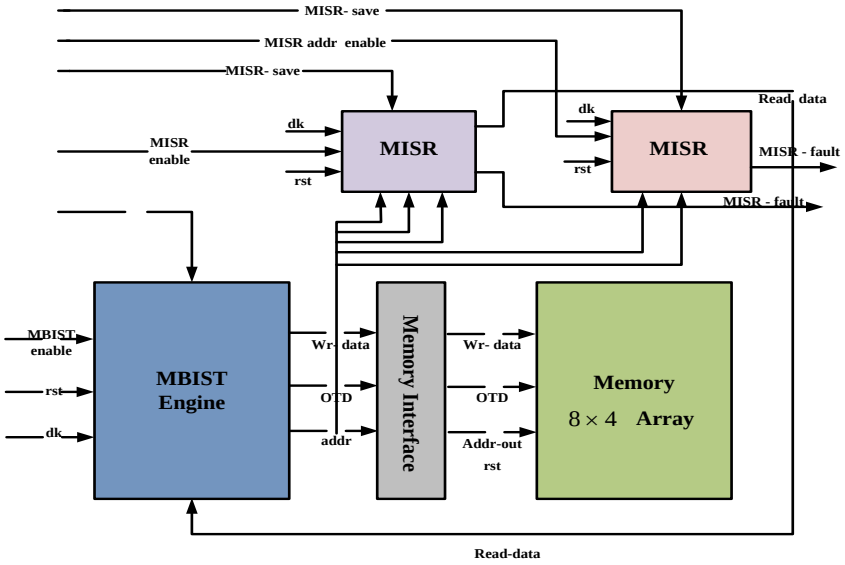


Figure 2. Proposed MBIST engine for faults defects in memory cells

2.3. Detection and classification of faults utilizing temporal channel reconfiguration multi-graph convolution networks

In this section, TCRMCN is discussed for detects and classifies the faults like write destructive fault, transition coupling faults, static coupling faults, stuck fault, coupling fault, read destructive fault, deceptive read destructive fault, disturb cell coupling faults, incorrect read fault, and idempotent coupling fault in the array. The incorporation of temporal channels improves the techniques capability to analyze fault progression the first layer of TCRMCN employs multi-channel convolution layers, where the temporal graph convolution operation utilizes an ordinary convolution mechanism as represented in (1):

$$k_{out} = \delta(\sum_{F}^{f=1} \hat{H} k_{in} + k_{in}) \varpi_f \quad (1)$$

Where, k_{out} denotes the generate matrices of different spatial regions, δ signifies the variations of the data, $\hat{H} k_{in}$ indicates the weight parameter, k_{in} designates the various features from the data, and ϖ_f denotes the learnable parameter. In the residual blocks in deeper layers enhance fault classification by preserving crucial features while enabling complex representation learning. Max pooling refines feature extraction by reducing dimensionality, while activation introduces non-linearity for improved reparability is given in (2):

$$Q_n = relu\{conv[q(w_n)]\} \quad (2)$$

Where, Q_n signifies the residual blocks of the layer, q indicates the pooling operation, and w_n denotes the quality of the array. Data wise information is extracted from memory arrays using multi-channel convolution layers to capture distinguishing features aiding in fault type differentiation. These features are further refined through residual blocks and pooling layers, enhancing memory fault representation from memory arrays is expressed in (3):

$$Y_s = \frac{1}{B \times V} \sum_{B}^{a=1} \sum_{V}^{b=1} k_{out}(a, b) \quad (3)$$

Where, Y_s denotes the scalable function, B signifies the features of layer, V indicates the activation function of layer, and (a, b) represent the direction factors from the memory arrays. The final layers of TCRMCN consist of fully connected layers that refined feature representations ensuring effective fault classification. The output layer employs the soft max activation function to fault detection process is represented in (4):

$$D_{out} = \beta_1 u_1 + \beta_2 u_{t1} \quad (4)$$

Where, D_{out} signifies the detection of faults, β represent the fully connected layers, and u_{t1} denotes the SoftMax activation function. Finally, the TCRMCN is employed for the detected and classified the faults as stuck fault, static coupling faults, write destructive fault, transition coupling faults, disturb cell coupling faults, coupling fault, read destructive fault, incorrect read fault, deceptive read destructive fault, and idempotent coupling fault in the array [12].

2.4. Optimization using multi-objective fitness dependent optimization algorithm

In this section MOFDOA is discussed for optimize BISR by efficiently selecting and allocating redundant memory cells enhancing fault correction and system reliability. BIST is a cost-effective integrated circuit embedded in SRAM memories to detect faults during read and write operations eliminating the need for expensive and time-consuming automated test equipment (ATE) [13]. Compared to previous optimization approaches, this optimizer works effectively and rapidly, resulting to fault repair and system dependability in less time. The MOFDOA improves performance when used to optimize BISR.

2.4.1. Stepwise procedure for multi-objective fitness dependent optimization algorithm

The step-by-step procedure is distinct to get the ideal value of (CAGNN) based on MOFDOA. Initially, MOFDOA distributes the population equally to optimize performance, and the best solution is promoted using the MOFDOA algorithm for enhanced efficiency. BIST is integrated and write operations while ensuring reliable fault detection and correction.

Step 1: initialization

During the initialization phase of the MOFDOA algorithm, a random population of candidate solutions is generated. The objective of this phase is to ensure a diverse set of starting points, allowing for efficient exploration of different regions within the solution space. This is represented in (5):

$$E_{u,h+1} = E_{u,h} + pace \tag{5}$$

where, E depicts the random starting population, u represent current individual number, h shows current iteration, and $pace$ representing movement rate and direction.

Step 2: random generation

The input created at arbitrary following start up. Ideal fitness values were determined rely on obvious BISR scenario.

Step 3: fitness function

Initialized BISR evaluations are employed to generate arbitrary solutions. The fitness function is evaluated for optimizing the data is accessed through spare memory value. It is expressed by using (6):

$$Fitness\ Function = optimize\ (\text{BISR}) \tag{6}$$

Step 4: pareto dominance

MOFDOA utilizes pareto dominance to efficiently select optimal solutions by balancing multiple objectives without compromising performance. It identifies non-dominated solutions can be improved without degrading another, ensuring an optimal trade-off. This approach enhances fault correction and system reliability in BISR is expressed in (7):

$$N_{Cu}(k_h) = N_u(k_h) + \widehat{K}\delta h_{max} \tag{7}$$

Here, N_{Cu} specifies new solution of circuit, $\widehat{K}$ represents the constant value, $N_u(k_h)$ represents the current solution of circuit, δh_{max} and depicts maximum perturbation acceptable between original mutated solutions.

Step 5: population update

The memory update procedure is crucial for BISR's ability to enhance fault correction and ensure system reliability. During this process, faulty memory cells are progressively replaced by redundant cells with MOFDOA improving the selection and allocation for optimal repair. This is given in (8):

$$IGD = \frac{\sqrt{\sum_{u=1}^{j} q_u^2 + Y_{TM} + D}}{x} \tag{8}$$

Where, IGD depicts the faulty memory cells, D represents the redundant cells, x shows true pareto optimal solutions, and q_u^2 represent Euclidean distance among closest obtained pareto optimal solutions.

Step 6: termination

The MOFDOA verification continues until it attains a null state, maximizing spare memory allocation depending upon defective cells. The proposed approach ensures efficient memory testing and fault repair through control flow. This optimizes BISR performance by dynamically adjusting the repair process based on fault patterns effectively [14]–[16].

3. RESULTS AND DISCUSSION

In this part, the investigational outcomes of the indicated procedures are discussed. An experimental and outcome are reached using mentor graphics and Questa simulator. This study focused on a 64-bit Windows operating system. The obtained outcome of the proposed TCRMCN-MBIST-BISR-SOC method is analyzed with the existing techniques like ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively [17], [18].

3.1. Performance measures

Accuracy, delay, access time, and power consumption evaluation measures are employed to explain the performance of proposed method.

3.1.1. Accuracy

It is the proportion of the overall count of forecasts produced for a dataset to the number of exact forecasts. It is measured through in (9):

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (9)$$

Here, TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative.

3.1.2. Delay

The period needed to be transmitted and receive a packet. It is computed as seconds and it is computed as given in (10):

$$delay = T_S - T_R \quad (10)$$

Here, T_S represents the message transmitting time and T_R represents the message receiving time.

3.1.3. Access time

Gaining accesstime is the duration required for a computer system or storage device to retrieve data from storage or memory. It is calculated as given in (11):

$$AccessTime = seektime + latencytime + transfertime \quad (11)$$

3.1.4. Power consumption

Power consumption is the amount of energy employed for each unit of time. In digital systems, power utilization is quite significant. It shortens the battery life of computers and smartphones, among other portable electronics. It is calculated as given in (12):

$$q_b = \mu T_{vol}^2 \left(M_d . D_g\right) \quad (12)$$

where, D_g signifies frequency of clock, M_d implies capacitance of load, μ denotes activation factor, and T_{vol}^2 indicates supply voltage.

3.2. Experimental and simulation approach

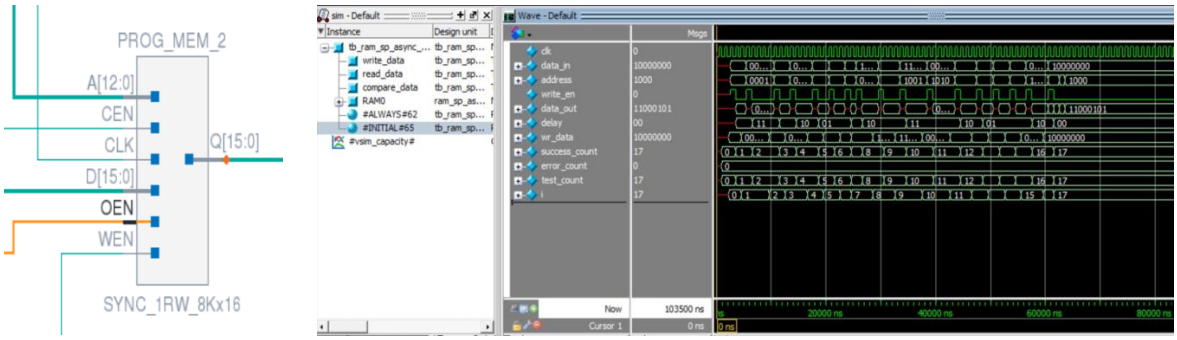
Embeddedmemories are tested with BIST and BISR following steps using TCRMCN-MBIST-BISR-SOC algorithm approach shown in Table 1.

The proposed TCRMCN-MBIST-BISR-SOC algorithm yields the injected faults and creates test patterns. Here, the numbers of faulty cells are detected through the TCRMCN-MBIST-BISR-SOC algorithm for assigning spare rows and columns. Based on the clock and address signals data will be written and read according to algorithm specified in SOC design (Figure 3). Schematic view of memory writes, read, address, data, and output signals shown in Figure 3(a). Simulation outcomes for controller functional checking where read write operations and defect kind detection are displayed in Figure 3(b). with the help of Questa

simulator tool. The writing and reading data patterns within the memories units are represented by this simulated waveform. The writing and reading tasks were completed in accordance with proposed algorithm approach. The fault table for BIRA computes and writes and reads checks of all fault types. In this experimental approach it is used to test the SRAM 16×8-bit words. Data write is active on the positive edge of clock if write enable equal to one (synchronous write), read is done 1 clock after the address is changed (synchronous read) based on proposed algorithm stuck, coupling, read destructive, write destructive, transition, static, disturb cell coupling, incorrect, deceptive read destructive, idempotent faults will check, and compared with other methods, details are mentioned in performance analysis [19].

Table 1. Steps for testing embedded memories with BIST and BISR

Step	Action	Description
1	Write data	Write the data to the storage position specified by the address.
2	Address	Specifies the address of the area in the memory where the memory's data will be accessed.
3	Write	To write to memory, use the writes-enabled signaling.
4	Read	The read enable signal indicates that the memory is being read.
5	Read data	The read data bus holds the read data from a specified memory address.
6	Write 0	To the memory address, write the logic value '0'.
7	Write 1	To the memory address, write the logic value '1'.
8	Read 0	Read the storage cell's logic value '0'.
9	Read 1	Read the storage cell's logic value '1'.



(a) (b)

Figure 3. Schematic view of memory; (a) SRAM bit cell schematic and (b) simulation waveforms for memory access operations

3.3. Performance analysis

The following data is generated using Questa simulator tool by changing the algorithms. Based on the algorithm and faults will be injected and control in memory and tested with BISR optimization.

Figure 4 displays the evaluation of accuracy performance. TCRMCN ensures more accurate defect detection by allowing dynamic response throughout testing. Optimization-based BISR ensures effective fault correction, lowering undetected faults, while MBIST increases the accuracy of memory testing. Here, TCRMCN-MBIST-BISR-SOC attains 9.28%, 8.78%, and 9.29% higher accuracy at stuck fault; 7.54%, 6.54%, and 9.42% higher accuracy at coupling fault, 5.96%, 7.25%, and 8.24% higher accuracy at read destructive fault, 5.27%, 9.34%, and 8.23% higher accuracy at write destructive fault, 7.36%, 8.31%, and 9.12% higher accuracy at transition coupling faults, 7.12%, 8.16%, and 9.75% higher accuracy at static coupling faults, 7.45%, 8.45%, and 6.56% higher accuracy at disturb cell coupling faults, 6.45%, 7.12%, and 8.36% higher accuracy at incorrect read fault, 6.34%, 7.56%, and 9.85% higher accuracy at deceptive read destructive fault, and 5.78%, 8.63%, and 9.78% higher accuracy at idempotent coupling fault accuracy at disease when comparing to the existing ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively.

Figure 5 portrays the evaluation of delay performance. By verifying that resources are distributed dynamically according to current demands, TCRMCN reconfiguration helps to minimize delays. The system's low delay is largely achieved by this mix of parallelism and effective resource management. Here, TCRMCN-MBIST-BISR-SOC attains 9.45%, 5.36%, and 8.28% lower delay at stuck fault; 5.14%, 8.15%, and 5.19% lower delay at coupling fault, 8.56%, 6.36%, and 5.78% lower delay at read destructive fault, 6.39%, 8.36%, and 7.58% lower delay at write destructive fault, 8.36%, 7.58%, and 7.69% lower delay at

transition coupling faults, 6.96%, 8.97%, and 7.58% lower delay at static coupling faults, 9.36%, 7.54%, and 5.67% lower delay at disturb cell coupling faults, 7.58%, 8.26%, and 6.39% lower delay at incorrect read fault, 6.69%, 8.58%, and 9.8% lower delay at deceptive read destructive fault, and 7.58%, 7.12%, and 8.57% lower delay at idempotent coupling fault when comparing to the existing ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively.

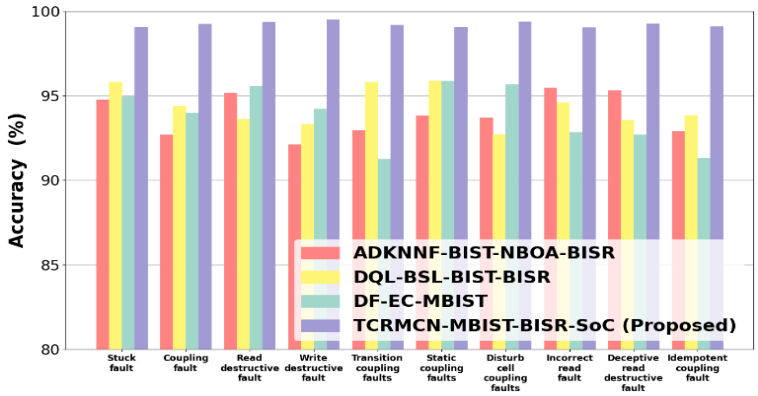


Figure 4. Evaluation of accuracy

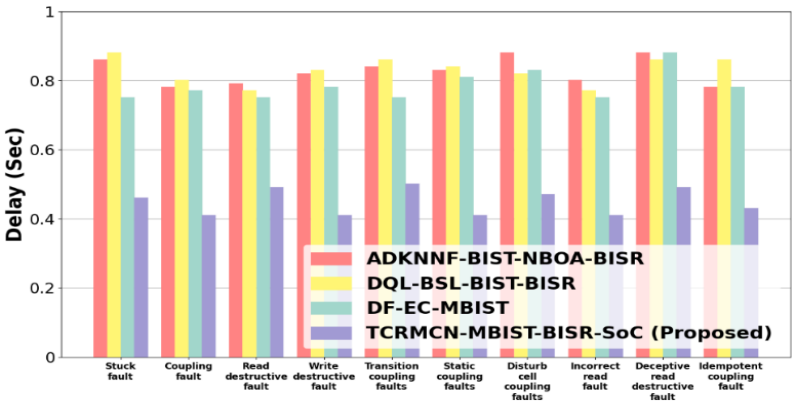


Figure 5. Performance analysis of delay

Figure 6 portrays the evaluation of access time performance. The BISR mechanism uses the multi-objective MFOA, which priorities performance metric including power consumption, and delay while minimizing overheads. This has a direct effect on access time by facilitating faster and more effective testing and repair procedures. Here, TCRMCN-MBIST-BISR-SOC attains 8.96%, 5.45%, and 9.89% lower access time at stuck fault; and 6.98%, 6.97%, and 4.31% lower access time at coupling fault, 9.57%, 6.36%, and 5.69% lower access time at read destructive fault, 7.25%, 8.57%, and 7.89% lower access time at write destructive fault, 6.37%, 8.57%, and 9.32% lower access time at transition coupling faults, 6.22%, 8.56%, and 8.75% lower access time at static coupling faults, 6.55%, 8.15%, and 7.58% lower access time at disturb cell coupling faults, 7.58%, 7.69%, and 7.38% lower access time at incorrect read fault, 9.54%, 8.86%, and 7.88% lower access time at deceptive read destructive fault, and 9.58%, 7.83%, and 8.98% lower access time at idempotent coupling fault when comparing to the existing as ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively.

Figure 7 portrays the evaluation of power consumption performance. Utilizing temporal channel reconfiguration reduces energy waste by enabling dynamic power usage adaptability based on system activity. A MOFDOA is used to combine power consumption with other design objectives. Here, TCRMCN-MBIST-BISR-SOC attains 8.69%, 8.36%, and 7.98% lower power consumption at stuck fault, 5.69%, 8.12%, and 9.42% lower power consumption at coupling fault, 8.69%, 5.23%, and 6.36% lower power consumption at read destructive fault, 6.69%, 8.24%, and 7.26% lower power consumption at write

destructive fault, 6.63%, 7.13%, and 8.25% lower power consumption at transition coupling faults, 6.54%, 7.58%, and 9.78% lower power consumption at static coupling faults, 7.75%, 7.75%, and 5.56% lower power consumption at disturb cell coupling faults, 6.45%, 6.69%, and 8.69% lower power consumption at incorrect read fault, 7.25%, 5.56%, and 8.55% lower power consumption at deceptive read destructive fault, and 5.68%, 8.53%, and 9.68% lower power consumption at idempotent coupling fault when analyzing to the existing ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively.

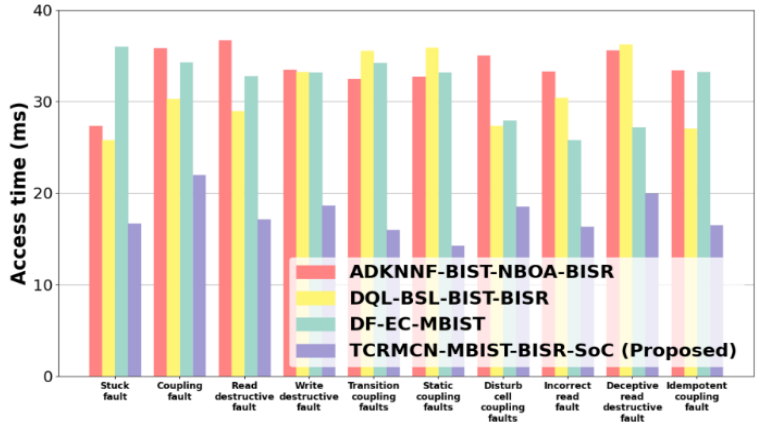


Figure 6. Evaluation of access time

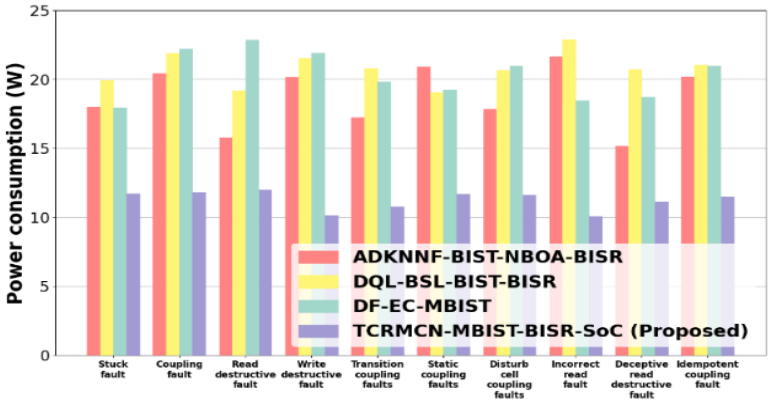


Figure 7. Evaluation of power consumption

Figure 8 depicts the training and validation accuracy vs epoch analysis, showing the learning progression of the model during the fault detection and classification task in the SRAM memory array. The training accuracy increases steadily, indicating the model's improved performance on the training data. The validation accuracy rises initially, demonstrating good generalization to unseen data in the early epochs. However, it later plateaus or slightly decreases, proposed the onset of over fitting where the model begins to learn noise and specific patterns from the training data instead of generalizable features. To address this, techniques like early stopping, regularization, and data augmentation can be applied to enhance the model's generalization and maintain robust performance in fault detection and classification [20].

Figure 9 depicts the loss vs epoch analysis, illustrating the optimization process of the model during the fault detection and classification task in the SRAM memory arrays the training loss gradually drops, the model is successfully learning and reducing errors on the training set. Similarly, the validation loss initially decreases, demonstrating good generalization to unseen data. However, beginning of over fitting, in which the model learns patterns unique to the training data rather than generalizable features, is indicated if the validation loss begins to plateau or rise while the training loss keeps reducing. To mitigate this, techniques like early stopping, dropout, weight regularization, and data augmentation can be employed to improve the model's robustness and prevent over fitting. Table 2 displays the Benchmark table for literature review [21].

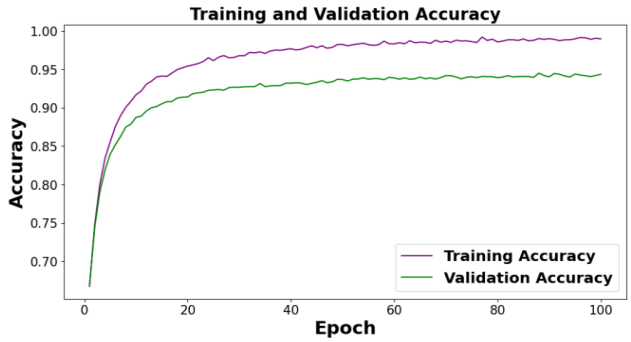


Figure 8. Performance analysis of training and validation accuracy vs epoch

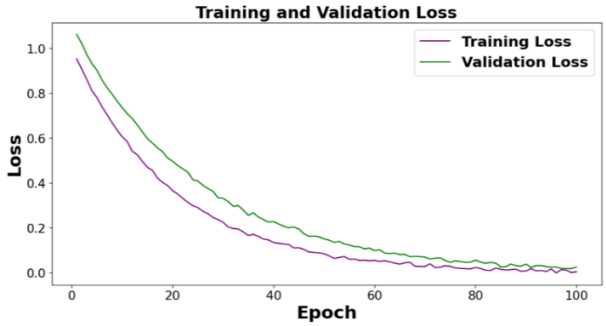


Figure 9. Performance analysis of loss vs epoch

Table 2. Methods and parameters comparison metrics

	Accuracy	Delay time	Access time	Power consumption
TCRMCN-MBIST-BISR-SOC	High	Low	Low	Low
ADKNNF-BIST-NBOA-BISR	High	High	High	High
DQL-BSL-BIST-BISR	Low	High	High	High
DF-EC-MBIST	Low	High	High	High

Table 2 presents the benchmark analysis for the literature review, evaluating the performance of various methods in SOC devices. The study investigates the effectiveness of the TCRMCN-based MBIST with Optimization-based BISR TCRMCN-MBIST-BISR-SOC approach in edge computing environments. Compared to existing methods, the proposed TCRMCN-MBIST-BISR-SOC model demonstrates superior accuracy, significantly reducing detection errors while optimizing power consumption and access time. Compared to existing methods, the proposed TCRMCN-MBIST-BISR-SOC model demonstrates superior accuracy 99.05%, significantly reducing detection errors while optimizing power consumption 11.67 W and access time 16.63 ms. The model also achieves a low delay of 0.46 seconds, outperforming conventional approaches, which have accuracy ranging from 88.43% to 95.77%, access times between 25.76 ms and 35.95 ms, and power consumption between 17.91 W and 22.45 W. Underscoring its effectiveness in real-time SOC environments. These results highlight the robustness of TCRMCN-MBIST-BISR-SOC, making it a promising solution for secure and efficient data processing in system-on-chip devices [22].

The proposed method, TCRMCN-MBIST-BISR-SOC, demonstrates higher accuracy, lower delay time, reduced access time, and low power consumption compared to other existing methods, such as ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST. A detailed comparative analysis is provided through the graphs and tables presented below, highlighting the performance improvements achieved by the proposed approach [23]-[25]. Comparisons of accuracy, delay time, access time, and power consumption with various algorithms are mentioned in Table 2.

4. CONCLUSION

The proposed TCRMCN-MBIST-BISR-SOC method focuses on MBIST based on SOC. MBIST check for defects in the memory cells. The fault is detected and classified using TCRMCN and the data is

passed to BISR, where the faulty memory cells are replaced using redundant memory cells and it is optimized by using MOFDOA. The proposed TCRMCN-MBIST-BISR-SOC method is executed in Questa simulator. The effectiveness of the proposed TCRMCN-MBIST-BISR-SOC approach contains 8.96%, 5.45%, and 9.89% lower access time and 8.69%, 8.36%, and 7.98% lower power consumption when analyzed to the existing techniques such as ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST respectively. Focused proposed study TCRMCN-MBIST-BISR-SOC method open up many research tracks where minimize power consumption, decrease delay time, boost accuracy, and shorten access time, also less implementation cost and are overhead. compared with other methods ADKNNF-BIST-NBOA-BISR, DQL-BSL-BIST-BISR, and DF-EC-MBIST.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Madhava Rao Jillella	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Prashanth	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	
Narayanappa Ananda														

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing -**O**riginal Draft
E : Writing - Review &**E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable. New data is created for this paper.

REFERENCE

[1] N. Lepri, A. Glukhov, L. Cattaneo, M. Farronato, P. Mannocci, and D. Ielmini, "In-Memory Computing for Machine Learning and Deep Learning," *IEEE Journal of the Electron Devices Society*, vol. 11, pp. 587–601, 2023, doi: 10.1109/JEDS.2023.3265875.

[2] E. Rama *et al.*, "Trustworthy Integrated Circuits: From Safety to Security and Beyond," *IEEE Access*, vol. 12, pp. 69603–69632, 2024, doi: 10.1109/ACCESS.2024.3400685.

[3] H. Lee, J. Kim, J. Park, and S. Kang, "STRAIT: Self-Test and Self-Recovery for AI Accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 9, pp. 3092–3104, 2023, doi: 10.1109/TCAD.2023.3236875.

[4] C. Thangam and R. Manjith, "Advancing Low Power BIST Architecture with GAN-Driven Test Pattern Optimization," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 40, no. 6, pp. 691–705, 2024, doi: 10.1007/s10836-024-06153-3.

[5] S. Alnatheer and M. A. Ahmed, "ADKNN fostered BIST with Namib Beetle optimization algorithm espoused BISR for SoC-based devices," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 1, pp. 90–101, 2024, doi: 10.11591/ijeecs.v35.i1.pp90-101.

[6] M. A. Ahmed and S. Alnatheer, "Deep Q-Learning with Bit-Swapping-Based Linear Feedback Shift Register fostered Built-in Self-Test and Built-in Self-Repair for SRAM," *Micromachines*, vol. 13, no. 6, 2022, doi: 10.3390/mi13060971.

[7] P. Bharati, D. Singh, and A. Singh, "Design a fast and energy efficient MBIST architecture using Verilog," *Hybrid and Advanced Technologies*, pp. 212–217, 2025, doi: 10.1201/9781003559139-28.

[8] J. Yoon, H. Lee, Y. Moon, S. H. Shin, and S. Kang, "A Built-In Self-Repair With Maximum Fault Collection and Fast Analysis Method for HBM," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 5, pp. 2014–2025, 2025, doi: 10.1109/TCAD.2024.3499903.

[9] F. Angione *et al.*, "A Toolchain to Quantify Burn-In Stress Effectiveness on Large Automotive System-on-Chips," *IEEE Access*, vol. 11, pp. 105655–105676, 2023, doi: 10.1109/ACCESS.2023.3316511.

[10] S. Lei, B. Tang, Y. Chen, M. Zhao, Y. Xu, and Z. Long, "Temporal channel reconfiguration multi-graph convolution network for skeleton-based action recognition," *IET Computer Vision*, vol. 18, no. 6, pp. 813–825, 2024, doi: 10.1049/cvi2.12279.
[11] J. M. Abdullah, T. A. Rashid, B. B. Maaroof, and S. Mirjalili, "Multi-objective fitness-dependent optimizer algorithm," *Neural Computing and Applications*, vol. 35, no. 16, pp. 11969–11987, 2023, doi: 10.1007/s00521-023-08332-3.
[12] S. Zhang, Z. Wang, and Y. Wang, "The MBIST Implementation of SoC Chip," *2024 9th International Conference on Integrated Circuits and Microsystems, ICICM 2024*, pp. 73–77, 2024, doi: 10.1109/ICICM63644.2024.10814273.
[13] S. V. Jois and H. V. R. Aradhya, "Robust and Efficient Implementation of Design for Testability in Integrated Circuits," in *8th IEEE International Conference on Computational System and Information Technology for Sustainable Solutions, CSITSS 2024*, Nov. 2024, pp. 1–5, doi: 10.1109/CSITSS64042.2024.10816981.
[14] Khushi and K. Singh, "Performance Analysis of March M & B Algorithms for Memory Built-In Self-Test (BIST)," in *Proceedings - 2022 IEEE World Conference on Applied Intelligence and Computing, AIC 2022*, 2022, pp. 78–84, doi: 10.1109/AIC55036.2022.9848869.
[15] L. W. Ying, R. Hussin, N. Ahmad, L. W. Fook, and A. Z. Jidin, "Modified March MSS for Unlinked Dynamic Faults Detection," in *2022 IEEE 20th Student Conference on Research and Development, SCOReD 2022*, 2022, pp. 68–72, doi: 10.1109/SCOReD57082.2022.9974097.
[16] H. Wu, "MBIST Repair Mechanism and Implementation," in *2022 China Semiconductor Technology International Conference, CSTIC 2022*, 2022, doi: 10.1109/CSTIC55103.2022.9856843.
[17] R. Manasa, R. Verma, and D. Koppad, "Implementation of BIST Technology using March-LR Algorithm," in *2019 4th IEEE International Conference on Recent Trends on Electronics, Information, Communication and Technology, RTEICT 2019 - Proceedings*, 2019, pp. 1208–1212, doi: 10.1109/RTEICT46194.2019.9016784.
[18] R. Kademani, S. Shirol, S. Siddamal, H. M. Vijay, and M. Rajeshwari, "GUI Implementation of Logic Built in Self-Test[LBIST]," in *2024 3rd International Conference for Innovation in Technology, INOCON*, 2024, doi: 10.1109/INOCON60754.2024.10511846.
[19] V. Midasala, G. Lakshminarayana, V. P. C. Reddy, B. H. Krishna, P. M. Kumar, and N. V. Krishna, "Design of Hybrid Memory Built in Self Test using Linear Feedback Shift Registers," in *6th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2022 - Proceedings*, 2022, pp. 564–568, doi: 10.1109/ICECA55336.2022.10009449.
[20] S. B. Ghale and P. Namita, "Design and Implementation of Memory BIST for Hybrid Cache Architecture," in *Proceedings of the 6th International Conference on Communication and Electronics Systems, ICCES 2021*, 2021, pp. 26–31, doi: 10.1109/ICCES51350.2021.9489225.
[21] A. Singh, G. M. Kumar, and A. Aasti, "Controller Architecture for Memory BIST Algorithms," in *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2020, doi: 10.1109/SCEECS48394.2020.43.
[22] V. Govindaraj and S. Dhanasekar, "Test Power Reduction through Reordering Algorithm Implementation and Advancements in BIST Architecture," *IETE Journal of Research*, vol. 70, no. 9, pp. 7513-7525, 2024, doi: 10.1080/03772063.2024.2352146.
[23] Y. Ryu *et al.*, "A 16 GB 1024 GB/s HBM3 DRAM with On-Die Error Control Scheme for Enhanced RAS Features," in *Digest of Technical Papers - Symposium on VLSI Technology*, 2022, vol. 2022, pp. 130–131, doi: 10.1109/VLSITechnologyandCir46769.2022.9830391.
[24] A. Paul and P. R. Antony, "Optimized microcode BIST architecture for multiple memory cores in SoCs," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings*, 2018, pp. 910–914, doi: 10.1109/RTEICT42901.2018.9012276.
[25] Y. J. M. Shirur, H. R. Lakshmi, and V. S. Chakravarthi, "Implementation of Area Efficient Hybrid MBIST for Memory Clusters in Asynchronous SoC," in *Proceedings - 2014 5th International Symposium on Electronic System Design (ISED)*, 2014, pp. 226–227, doi: 10.1109/ISED.2014.57.

BIOGRAPHIES OF AUTHORS

Madhava Rao Jillella received the B.Tech. degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, Hyderabad in the year 2006 and M.Tech. in Power Electronics from Jawaharlal Nehru Technological University, Anantapuram in the year 2011. His research interests include VLSI design and power electronics. Currently he is pursuing Ph.D. from Visvesvaraya Technological University, Belgum, Karnataka, Under the guidance of Dr. Prashanths N A. Associate Professor, Department of EEE, BMS Institute of Technology and Management, Doddaballapura Main Road, Avalahalli, Yelahanka, Bengaluru 560064. He can be contacted at email: madhava.jmr1@gmail.com.

Dr. Prashanth Narayanappa Ananda received the B.E. degree in Electrical and Electronics Engineering from Visvesvaraya Technological University and M.Tech. in Computer Applications in Industrial Drives from Visvesvaraya Technological University in the year 2002 and 2005 respectively and Ph.D. from Jawaharlal Nehru Technological University, Anantapuram in 2020. He is currently working as Associate Professor in BMS Institute of Technology & Management, Bangalore. His research interests include wind energy system and FACTs controllers. He can be contacted at email: prashanthna@bmsit.in.