iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VI Month of publication: June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35640

Automated Safety & Security Systems for Industrial Application using Self-Healing Approach

B Sri SaiCharan[1], M Shashank[2], K Sumanth krishna[4], V RamaKrishna Sharma[4]
[1, 2, 3]*Student,* [4]*Associate Professor Department of Electronic and Communication Engineering, Sreenidhi Institute of Science and Technology, Ghatkesar, Telangana, India*

Abstract: The concept of self-healing for hardware systems is studied in this study, and a new approach is offered. In the way they offer healing and recovery abilities, key technologies have been offering imitations to biological entities. Digital systems with inspired uniform architecture offer better fault-tolerance capabilities. The ability of a system to detect and repair flaws or failures is known as self-healing. Given that current self-healing systems are focused on redundancy and spare blocks, one of the key issues with current self-healing approaches is area overhead and scalability for complicated structures. This research proposes a new method for self-healing based on embryonic hardware that does not require the use of spare cells. When compared to alternative techniques that rely on spare cells, the area overhead is reduced. The suggested method entails time multiplexing two functions in a single cell inside a single clock cycle. The proposed technique's reliability is investigated and compared to that of a traditional system with varying failure rates. This method can heal up to 50% of the cells, and each cell can only cover one neighbouring failed cell at a time. The proposed solution has a 9 percent area overhead, which is significantly lower than prior spare cell approaches. The proposed method is tested on two case studies: an ALU array and a neural network.

Keywords: Self-Healing, Embryonic hardware, Fault tolerant, Reliable

I. INTRODUCTION

These days, the electronic circuit industry is expanding in intricacy quickly with new gadgets and designs of equipment frameworks. Equipment disappointments occurring - while running some continuous undertakings are basic during activity for warming, maturing or general climate boundaries. Such disappointments will be a block to the framework to work appropriately. Self-mending is a critical methodology for dependable electronic frameworks that run under brutal conditions, for example, vast radiation and savage temperature in space applications. Self-mending is done by means of numerous methodologies and large numbers of them are in view of repetition which depends on recuperating defective cells also, the reintegration of them back into the framework. Self-fix is the substitution of defective cells by working extra cells. Self-recuperating term can be utilized for the two ideas [1] [2]. The part of self-recuperating is to recuperate the issue in the framework to keep up the activity with best and most significant length of time conceivable. A portion of oneself recuperating research in the circuit level is based on the circuit replication Where, the broken cell is supplanted by the extra one after the identification of a flawed cell by the controlling part. Self-mending can be accomplished in frameworks over various levels. The first is application-level recuperating [3], which is the capacity of an application, or a help, to recuperate itself. Next comes the System level mending which relies upon a programming language and configuration designs that we apply inside [4] [5]. At long last, the degree of equipment mending from engineering and down to circuit procedures is the least level of recuperating [6].

There are various strategies for self-recuperating like Dual Measured Redundancy (DMR) [7], Triple Modular Redundancy (TMR) [8], Embryonic Hardware EmHW [6], [9], too as different procedures like evolvable equipment and Intelligent Equipment System (IHW) structure that are planned with mending capacities as a primary concern [10]. DMR and TMR are two kinds of repetition methods. In DMR strategy, there are two indistinguishable occasions running in equal for similar application and their yields are associated with a comparator that triggers a confuse at whatever point an issue happens. TMR has three indistinguishable modules running in corresponding to give deficiency veiling by a larger part citizen at their yields to cover a sign issue. Embryonic equipment is an equipment framework with self-mending based on replication of little structure blocks with careful engineering also, it is motivated by the multicellular organic entities cell division also, separation components. The disadvantages of current methods are region and force utilization where they require repetitive equipment which is an additional overhead. EmHW is the same standard of self-recuperating component of multi-cell living beings for adaptation to non-critical failure equipment framework. Like multi-cell life forms in nature the design of circuits in view of EmHW is two-dimensional exhibit of electronic cells (e-cell) with equipment reconfiguration capacity.

E-cells have the capacities of versatile self (recuperating) in light of its structure. The construction is homogeneous where each cell can play out totally set of capacities and can be designed to work one of them as displayed in Fig. 1.

Each cell is designed to performs just a particular capacity that is chosen by the setup data in every cell. At the point when a cell comes up short the mindful module of self-finding triggers self-mending and the flawed cell will be supplant by the extra cell. Fig. 1 shows general cell structure, it is made out of six modules; I/O module, address module, setup module, control module, work module, and location module. The activity of every one is as per the following; I/O module is utilized for transmission signals between various cells. Address module decides quality data from its facilitate of area data what's more, decides the area of the cells. Design module stores the design data of the multitude of cells which copies DNA in organic cells. it additionally gives data for acknowledgment of self-mending. Control module controls all activities of the cell, for example, issue case, straightforward case, and inactive case. Capacity module is the preparing unit or the corresponding capacity which is controlled by the design data. Location module recognizes cell working status, if it's anything but an ordinary working condition or not.

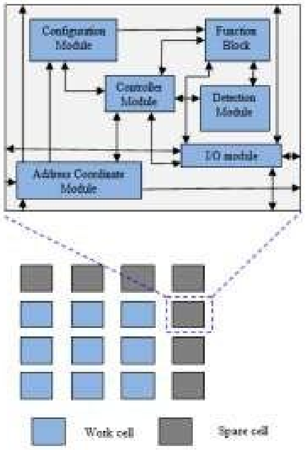


Fig1: Architecture on Embryonic Hardware

Self-mending isn't just pivotal for dependability but on the other hand it's one of the fundamental angles for a clever framework [10]. Other self-recuperating approaches roused from equipment insight ideal models incorporate evolvable equipment, where a framework can be developed from an underlying working point with mistakes or low wellness until it's anything but a path around defective cells through rearranging cells with hereditary programming. Another system can be viewed as where an online blend measure happens permitting the framework to track down another integrated design at whatever point an issue is distinguished. Such frameworks depend on a comparative uniform association like FPGAs or Genetic Programming units [11].

The remainder of the paper is coordinated as follows. Segment II presents proposed self-mending procedures. Segment III presents assessment lists on the proposed work. Execution and examination results are introduced in Sections IV followed by the end in Section V.

II. PROPOSED TECHNIQUE

The majority of self-mending approaches depend on excess by adding spare cells. At the point when a disappointment occurs, the broken cells are supplanted by the extra cells after recognition of deficiencies. Along with the referenced region overhead and adaptability, the planning of extra cells may create setback overhead after self-mending since the closest extra cell is exceptionally far away from the flawed one. For improved planning and framework's presentation, adding more extra cells will help yet on the opposite side the region will be higher. The region overhead can be decreased by adding less spare cells. Less extra cells prompt significant planning difficulties what's more, more unpredictable mending framework with expanding number of shortcomings after some time and entire framework disappointment or postponements. The proposed approach gives self-recuperating without adding spare cells which beat region overhead issues. The proposed approach regards every cell as an extra cell for its neighbour. Additionally, every cell can finish its undertaking and the assignment of the neighbour cell. The proposed method depends on time division multiplexing for self-mending. Every cell will be skilled to run two assignments inside a similar clock when a deficiency occurs.

One activity runs during the main portion of the clock cycle furthermore, the second during the other portion of the clock. On the off chance that the issue identification module identifies an issue then the neighbour of the broken one will run its errand alongside the undertaking of the flawed cell. The region overhead is exceptionally little contrasted with the past work. The planning issue gets like the instance of having an extra cell between every two neighbour cells which is a sensible association for planning.

The activity of the proposed procedure can be clarified as follows; When a shortcoming is recognized in a cell, the neighbour cell will get a control signal from regulator to run this defective cell task. The dynamic cell splits run time between its task and broken cells task utilizing a double edge set off "DET" cell. Thus, at the principal half of the clock cycle with the positive edge the dynamic cell run its unique assignment. In the second 50% of the clock cycle, it runs the assignment of the defective one. Fig. 2 shows the cell design of the proposed work. There are two information sources applied to DET cell and choice of one of them relies upon the worth of control signal (C) and clock (clk) esteem. When a deficiency occurs, the recognition block distinguishes issue area and the control block pulls the sign C up for this particular area. In this way, the DET cell drives typical information (I1) when clk esteem rises and chooses contribution of flawed cell (I2) when clk esteem drops as introduced in Fig. 3.

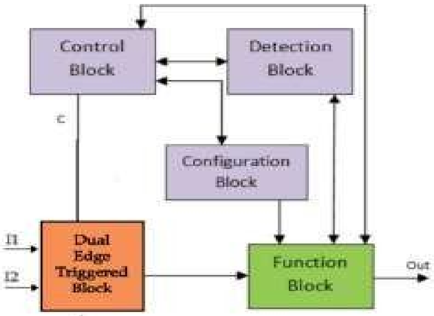


Fig2: Proposed cell Structure

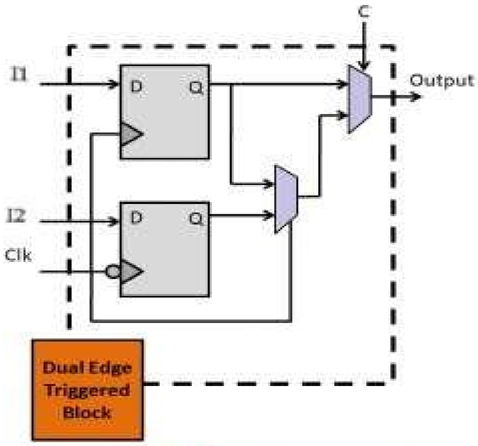


Fig3: Dual edge triggered circuit

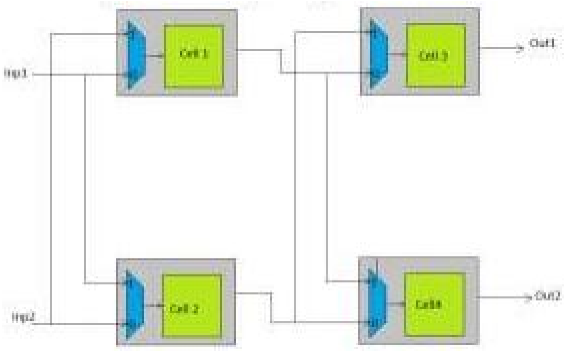


Fig4: Block diagram of the proposed technique

Fig. 4 shows a straightforward cluster to exhibit the thought. Accept there is a deficiency in cell 1, as per that the location block it's anything but a sign for the neighbours to check their initiation what's more, select one on them. Presently cell2 will remunerate the defective cell1 and control signal C1 will be one. Cell2 has two data sources applied to DET's information, one comes from the typical information and the subsequent which should go the contribution of cell1. The choice contribution of DET chooses one contribution to enter the cell 2 contingent upon the clock esteem. Cell2 has the activity of cell 1 undertaking to such an extent that when cell2 gets a control signal C1, the cell2 chooses the activity of cell 1 from setup module to be finished. For the negative worth of clock and whether C1 approaches 1 or 0 the DET block chooses inp2 and runs a capacity of cell 2. During the positive worth of clock and C1 approaches 1, the DET chooses inp1 and runs a component of cell 1. Moreover, the yield of cell 2 during the cycle season of cell 1's errand is taken care of two cell 3. A similar will occur on any cell where regular each cell has similar arrangement of capacities to be designed. Along these lines, rather than adding spare cells and increment region overhead, the self-recuperating is done on the framework utilizing the dynamic cells itself. On the off chance that there are numerous deficiencies another dynamic cell will redress faults as displayed in Fig. 5. Fig. 5 shows that there is issue in cell 2 and cell 9 and after self-mending association cell 6 will run errand of cell6 and undertaking of cell2. Additionally, cell 13 will run task of cell 13 and dead cell 9. The circumstance outline is displayed in Fig. 6, where for typical activity out1 is An and out2 is B. If there should be an occurrence of defective cell1, sign of flaw will ascend to one and cell two runs the activity of two cells as displayed in Fig. 6. Out1 is Z (coasting) and out2 is the yield of its undertaking (B) and yield errand of defective cell (A).

Throughput of the framework relies upon the clock activity of the framework. For the framework which depends on full cycle multiplexing where just one capacity works for every cycle, throughput will diminish to the half. The proposed technique in any case, keeps the cell which runs two assignments (its unique errand and defective cell's task) to work with double the pace of the typical activity. This cell will complete two jobs inside one clock cycle. Thus, the proposed approach works with a similar throughput with self-recuperating. This sort of self-recuperating has a place with the engineering level of shrewd equipment stack [10] where the checking occurs fair and square of e-cells and the regulator chooses to supplant a broken cell with one of the accessible close by cells in a multiplexing design as clarified. The reconfigurable texture here is equivalent to an EmHW based construction. This can be additionally reached out to utilize Genetic Programming units furthermore, advancing strategies for dynamic to improve the self-recuperating further.

Assessment Indexes for Proposed Embryonic Self-Healing

III. ASSESSMENT INDEXES FOR PROPOSED EMBRYONIC SELF-HEALING

Assessment list [12] for self-mending decides an assessment of self-recuperating procedure, examination and correlation on self-mending methodology. The assessment files which are shrouded in this paper are repetition rate, most extreme number of fixes, and self-mending time utilization.

A. Redundancy Rate

Redundancy rate is the proportion of the quantity of extra cells and the quantity of dynamic cells in early-stage cluster. Fashioner of undeveloped cluster needs to ensure that extra cells of the full exhibit of equipment assets are lesser. Additionally, ensure the capacity of self-mending of full cluster is higher. The proposed approach gives every cell as extra cell to its neighbour. Along these lines, the repetition rate is half without adding spare cells.

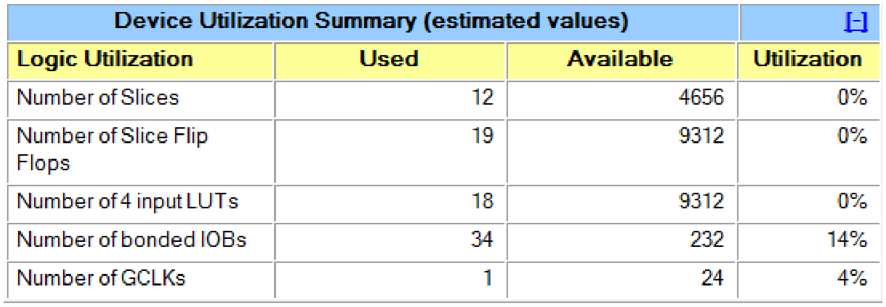
B. Maximum Number of Fixes

Maximum number of fixes alludes to the most extreme number of issue fix that can be fixed without help from anyone else mending technique in undeveloped cluster to return framework to ordinary working state. For the proposed method the greatest number of fixes is half of the complete number of cells in early-stage cluster.

C. Self-Healing Time Consuming

Self-recuperating time utilization is the time utilization from disappointment of cell to the issue fixed to recover the framework back to ordinary work. Thus, the maintenance speed is significant viewpoints in deficiency fixing stage. The proposed procedure depends on neighbour fixing which make the planning exceptionally straightforward doesn't take more deferral contrasted with the repetition strategy for self-mending.

The proposed strategy depends on neighbours for fixing which make the planning extremely basic and doesn't need steering delay contrasted with other excess procedures for self-recuperating. These highlights make the proposed approach an appealing methodology for self-mending.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	12	4656	0%
Number of Slice Flip Flops	19	9312	0%
Number of 4 input LUTs	18	9312	0%
Number of bonded IOBs	34	232	14%
Number of GCLKs	1	24	4%

Fig5: Design summary

IV. IMPLEMENTATION

The proposed technique relies on neighbours for repairing which make the mapping very simple and doesn't require routing delay compared to other redundancy techniques for self-healing. These features make the proposed approach an attractive approach for self-healing.

A. First Case Study: ALU Array

The modules are implemented using VHDL and the simulation results are obtained using ISE Xilinx 14.4 on vertex 5. Each cell includes ALU operation inside it and each one has a specific task that is determined by configuration block using the cell number Fig.1. Each cell includes ALU block operation and ROM to store the configuration of operation. In case of there is fault in any block, the neighbor one will compensate this fault using self-healing approach and configuration of each block.

B. Second Case Study: Neural Network

An Artificial Neural Network is based on a collection of nodes (neurons) and each connection between nodes can transmit a signal from one to another as shown in Fig.8. Each input is multiplied by a weight then send the result to the equivalent of a cell body. Using adder, the weighted signals is summed together to supply a node activation or activation function [13], [14]. The unit provides a high value output, if the activation function is higher than the threshold. The unit provides a low value output, if the activation function is lower than the threshold. Output of each neuron is calculated by:

$$R(t) = \sum_{i=k}^{m} C_m^i \exp\left(\frac{-\lambda i t}{2}\right)\left(1 - \exp\left(\frac{-\lambda t}{2}\right)\right)^{m-i} \quad (3)$$

Where X_j is the j^{th} neuron output in the proceeding layer, and n is the number of neurons, W_{ji} synaptic weight from j_{th} neuron to the i^{th} neuron in the proceeding layer, f is activation function, and b is bias.

Self-healing approach provides a healing to any faulty node in NN. If any node has fault, the neighbor node will run two function one is its function and the second one is the faulty node function. We implemented NN with input layer which has three nodes, three hidden layers with 4 nodes in each layer, and output layer with one node. This NN is used for classification between two different classes. NN is implemented using VHDL and the simulation results are obtained using ISE Xilinx 14.4 on vertex 5.

C. *Reliability*

The reliability is one of the important indicators for the system. The reliability is the ability of the system to perform its function within a certain period of time. In this paper the reliability analysis is considered. As many analyses are done in previous work on reliability [15], [16]. The probability of success for system is

$$p(t) = \exp(-\lambda t) \quad (2)$$

Where all units are identical and p(t) is assumed to be an exponential distribution failure and is the failure rate. Each cell can run two functions at same clock period for neighbor fault cell case then the reliability of the system is given by:

$$R(t) = \sum_{i=k}^{m} C_m^i \exp\left(\frac{-\lambda i t}{2}\right)\left(1 - \exp\left(\frac{-\lambda t}{2}\right)\right)^{m-i} \quad (3)$$

Where m is number of active unites for k function out from these unites.

V. ALGORITHM

This contains step by step implementation to get desired output.

A. open ISE
B. click on the new file and select new project
C. And give the file name to the new project and click on next, finish.
D. In the previous step we give the project name that name created the file. And select that file right click that file and select the copy of file
E. when select the copy of files and select the path of the code files and select all files and open that files.
F. The all-selected code files will be open on the ISE navigator
G. Now, we are selected to the implementation
H. And select the code and put on to set the top module of the code and check the syntax
I. When we completed the syntax and move to the RTL schematic. When it completed the RTL schematic will be open
J. when it completed to the implementation then move to the simulation
K. In that simulation we select the test bench of the code and click on behavioral model
L. When it completed then move to the simulate behavioral model when it completes the output window will be appeared.

VI. RESULTS

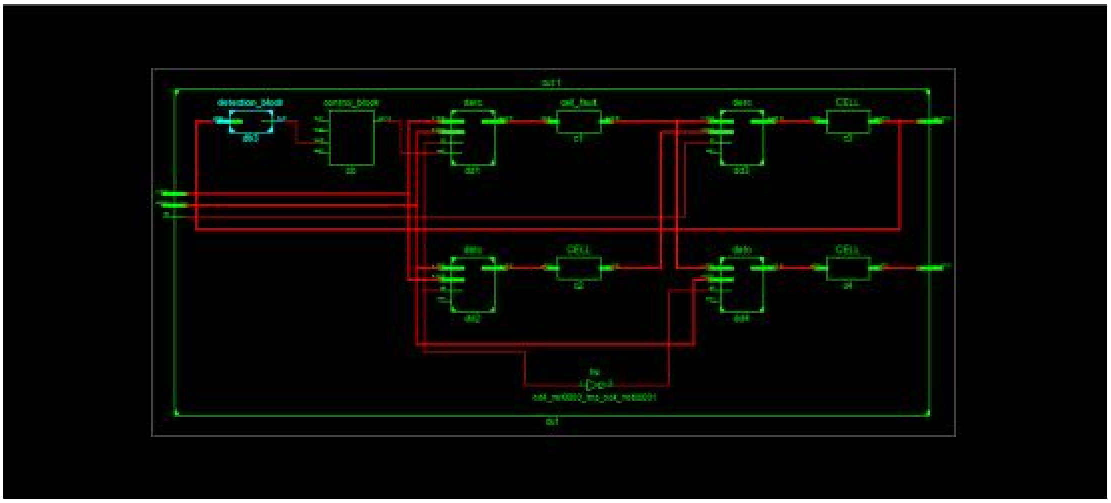
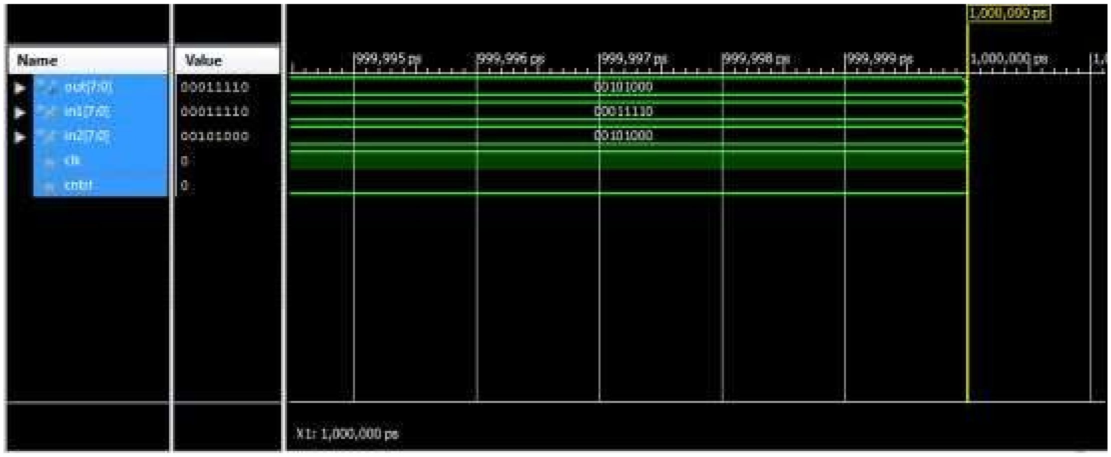
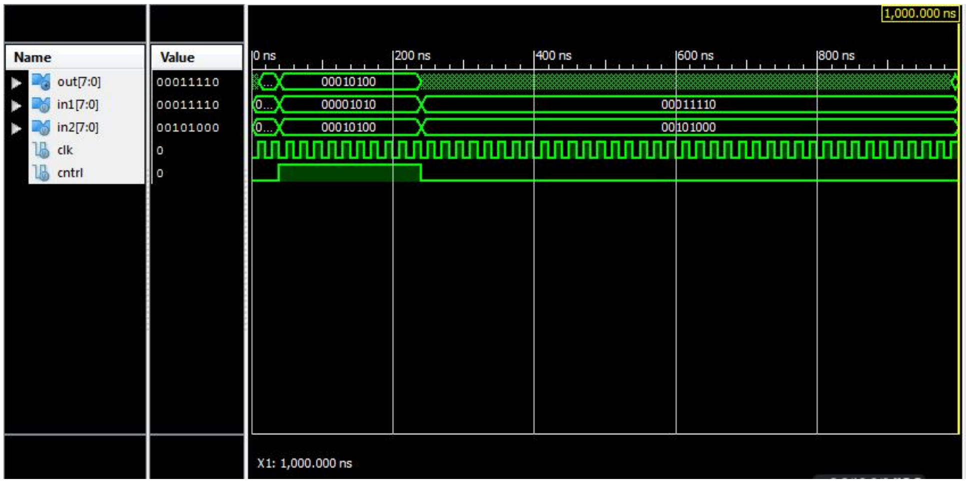


Fig6: RTL SCHEMATIC



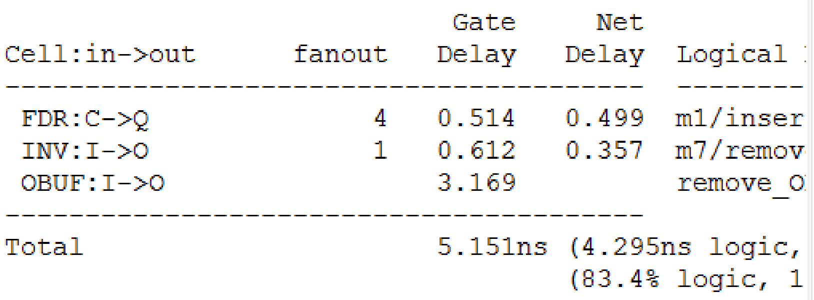
Name	Value
out[7:0]	00011110
in1[7:0]	00011110
in2[7:0]	00101000
clk	0
cntrl	0

Fig7: DETECTION



Name	Value
out[7:0]	00011110
in1[7:0]	00011110
in2[7:0]	00101000
clk	0
cntrl	0

Fig8: Output



```
                                    Gate     Net
Cell:in->out        fanout   Delay   Delay  Logical 
----------------------------------------  --------
 FDR:C->Q                4   0.514   0.499  m1/inser
 INV:I->O                1   0.612   0.357  m7/remov
 OBUF:I->O                   3.169          remove_O
----------------------------------------
Total                        5.151ns (4.295ns logic,
                                     (83.4% logic, 1
```

Fig9: Timing report

VII. CONCLUSIONS

In this we proposed a new technique for lowering the power loss during at speed test of sequential circuits with scan-based Logic Built in Self-Test using the scheme Lunch on Capture. The proposed solution allows designers to lower the probability that the delay induced by PD exhibited during at-speed test is erroneously interpreted as a delay fault, with consequent generation of a false test fail. We achieved this by lowering the AF of the CUT compared with conventional scan-based LBIST, by applying proper changes to the test vectors generated by the LFSR.

REFERENCES

[1] M. Furdek, M. Danko, P. Glavica, L. Wosinska, B. Mikac, N. Amaya, G. Zervas, and D. Simeonidou, “Efficient optical amplification in self-healing synthetic ROADMs,” in 2014 International Conference on Optical Network Design and Modeling, May 2014, pp. 150–155.

[2] S. M. Bowers, K. Sengupta, K. Dasgupta, B. D. Parker, and A. Ha jimiri, “Integrated self-healing for mm-wave power amplifiers,” IEEE Transactions on Microwave Theory and Techniques, vol. 61, no. 3, pp. 1301–1315, 2013.

[3] I. Ivan, C. Boja, and A. Zamfiroiu, “Self-healing for mobile applica tions,” Journal of Mobile, Embedded and Distributed Systems, vol. 4, no. 2, pp. 96–106, 2012.

[4] K. Khalil, O. Eldash, and M. Bayoumi, “Self-healing router architecture for reliable network-on-chips,” in IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2017.

[5] S. Narasimhan, S. Paul, R. S. Chakraborty, F. Wolff, C. Papachristou, D. J. Weyer, and S. Bhunia, “System level self-healing for parametric yield and reliability improvement under power bound,” in Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on. IEEE, 2010, pp. 52–58.

[6] M. R. Boesen, J. Madsen, and P. Pop, “Application-aware optimiza tion of redundant resources for the reconfigurable self-healing eDNA hardware architecture,” in 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), June 2011, pp. 66–73.

[7] Q.-Z. Zhou, X. Xie, J.-C. Nan, Y.-L. Xie, and S.-Y. Jiang, “Fault tol erant reconfigurable system with dual-module redundancy and dynamic reconfiguration,” vol. 9, no. 2, 2011, pp. 167–173.

[8] T. Koal, M. Ulbricht, P. Engelke, and H. T. Vierhaus, “On the feasibility of combining on-line-test and self repair for logic circuits,” in 2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), April 2013, pp. 187–192.

[9] K. Khalil, O. Eldash, and M. Bayoumi, “A novel approach towards less area overhead self-healing hardware systems,” in International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2017.

[10] O. Eldash, K. Khalil, and M. Bayoumi, “On on-chip intelligence paradigms,” in IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), April 2017.

[11] L. Shao, L. Liu, and X. Li, “Feature learning for image classification via multiobjective genetic programming,” IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 7, pp. 1359–1371, 2014.

[12] F. Meng, J. Cai, Y. Meng, S. Wu, and T. Wang, “Evaluation index system for embryonic self-healing strategy,” in 2016 International Conference on Integrated Circuits and Microsystems (ICICM), Nov 2016, pp. 86–90

[13] V. Vapnik and R. Izmailov, “Knowledge transfer in svm and neural networks,” Annals of Mathematics and Artificial Intelligence, vol. 81, no. 1-2, pp. 3–19, 2017.

[14] Z. Zhang, “Artificial neural network,” in Multivariate Time Series Analysis in Climate and Environmental Research. Springer, 2018, pp. 1–35.