International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-8)

Research Article **August 2017**

A Self-Repairing Digital System with High-Quality Scalability and Fault Coverage

[1]S. Ravichand, [2]T. Madhu, [3]M. Sailaja
[1] Research Scholar, JNTUK, Kakinada, Andhra Pradesh,
[2] Professor in ECE, Swarnandhra Inst. of Engg. and Tech., Narasapur, Andhra Pradesh, India
[3] Professor in ECE, JNTUK, Kakinada, Andhra Pradesh, India

Abstract-

In any fault tolerant or BIST system the primary goal is to covenant with faults that arise in the indented system. The proposed system using genetic algorithm to optimize the performance and area of given circuit. This approach is supple for combinational circuit design. The use of four spare cells simplifies the operation of the active block in the current system; it needs more space to establish itself so it is considered as overhead. The proposed method of fault detection and correction for logical errors using genetic algorithm decreases the area overhead. Detection of Fault in the memory unit through BIST implementation increases the speed but replacing the existing faulty block with fault free block degrades the fault analyzing capabilities. Utmost care has on all the works implemented for the process of minimizing the error in different digital process. Therefore, with the new scope of proposing the method of reducing the error flow for the application of medical field, aeronautical, satellite broadcasting is described very efficiently in this paper. The simulation results of the fault tolerant and self-repairing method using genetic algorithm is presented.

Keywords: Built in Self test, Fault detection, Active block, Genetic algorithm

I. INTRODUCTION

Electronic equipment used today has a great influence of integrated circuits, which occupy their place almost in device, digital VLSI circuits in specific. Digital VLSI/ULSI circuits used to design integrated circuits made all electronic commodities price slashed down to earth. Integration and scaling of transistors in larger number made a small chip into a micro level one. Thus integrated circuits have dramatically changed the design time, size of circuit or module. Mass production of these circuits in the form of integrated circuits yields more reliability, capability and more interestingly the design for replacement of common transistors.

Fault Detection, correction and tolerance has always been a crucial aspect when it comes to digital circuits. With a countless number of logical gates that continue to increase, it has pulled out to a virtually impossible state to fully test all input combination possible for circuit designs for testing purposes. During the process of manufacturing any digital devices if any defect is introduce that is said to be an error. A fault is said to be detected if a specific test pattern used with the primary inputs could detect the specific fault and contain differing primary output results from the original design.

High level fault modeling provides the ability to use simulation based design verification. At this fundamental level most common faults, popular faults models in the testing of digital arena like stuck-at faults, Bridging faults and delay faults occur to affect the performance of the device. The most versatile fault model for logical circuits testing is stuck-at fault model.

The difference occurred between designed values to the actual value i.e. between a theoretical design to the practical output. The deviation generated affects the performance and behavior of the designed system. So the designed circuit, system or a machine has exposed to its own methodologies of mitigating inexcusable causes of error, happened may or may not be intentional and might not be noticed for long time until the functionality for its right set comes into act to evaluate the originality. Error diagnosis plays an essential role in providing correct VLSI products.

Faults are understood to be an abnormal change of system function or defect at the component, equipment, or subsystem that may or may not lead to physical failure or breakdown. If faults occur, the outcome has the potential of being catastrophic by possibly endangering lives. It is imperative that uncovering the location of faults is critical. Some traditional approaches to fault diagnosis have been installing multiple sensors and hardware, analytical or functional redundancy, and a combination of hardware and analytical redundancy.

The purpose of this method is to implement the detection and diagnosing of faults within the circuit while under radiation. Attempting to diagnose the location of faults and errors on a circuit continues to be investigated thoroughly. It is implied that uncovering the location of faults is safety critical. Some conventional methods to diagnose fault have been installing many sensors and hardware, analytical and functional redundancy might be a combination of hardware

© www.ermt.net All Rights Reserved

redundancy. The basic purpose at these methods so as to implement the detection and diagnosing of faults within the circuit when projected to radiation.

1.1 Radiation Effects on Electronics

Operational reliability is one of the key principal concerns in microelectronic systems. This is particularly true of space bound systems since they are exposed to ionizing radiation and their operating conditions do not allow for quick and easy restoration of failed or malfunctioning components. The harsh space environment can cause severe damage and malfunction on unprotected electronics.

Long periods of time and exposure to space's callous energy particles can degrade even the best device's performance, leading to component failure. Everything from major components to the wiring and cabling of electronic devices can be seriously affected by radiation. This section will explore key components of radiation effects on electronics related to the research goals of this thesis. Three key issues are discussed further, total dose effect, single event effects (SEE), and single event upsets.

A low density parity check codes for error correction in Nano scale devices are more prone to soft errors than micro scale devices. Storage components built using Nano scale elements have order of magnitude higher rates of faults. Nano memories would also need to provide online error correcting codes (ECCs) for high soft error rates, to employ ECC with high fault rates. To achieve higher error-correcting capability, codes like Reed-Solomon or BCH require more sophisticated decoding algorithms. Complex operations would be difficult to implement using Nano scale PLAs. Soft error correction is critical for different Nano scale devices, performing storage, computation and communication error correction technique is used for nano scale memory actual prototype results with actual memory traces from different domains and injecting faults with different fault models. Memory size increases, area increases and overhead decreases.

An error detection and error correction technique for a transmission, chip fabrication is one of the crucial topics in the digital electronics. This technique fully performs its action when the information or the process is get read from the storage device. The information describes through literature survey is entirely get overcome by the error detection and error correction by the use of self-repairing method. Error correcting code, error detecting and correcting codes is already get implemented. But this performs its action by using hamming codes and low parity check codes. With the help of this technique fault-tolerant systems deal with only a single fault. The proposed technique has describes about the self – repairing method using genetic algorithm for effective design structure.

In future the simulation model designed for the fault tolerant or self -repairing system is utilized for the hardware implementation. And also to develop a compiler that can automatically synthesize the bit strings of the target circuits for an FPGA from VHSIC hardware description language codes by embedding the self-repairing architecture. In computer based analysis system, these processing techniques are used in order to facilitate and improve system and transmission efficient. It is hoped that the proposed self-repairing with two spare cells using genetic algorithm provide a more reliable and stable system.

II. GENETIC ALGORITHM

Biological evolutionary process has inspired greatly the selection of genetic algorithm, mutation and crossover. In huge and ambiguities search places probabilistic techniques are used by genetic algorithms (GA) to assess optimal solutions globally. "Chromosomes" reveal – population, a candidate solution in the process of Gas. A gene is referred to a bit present in a chromosome depending on the application fitness value is made associated with each chromosome. Where its competency rates it as a solution. Parent chromosome of the population occurs initially to start the process is generated very randomly. By crossovers there evolve offspring chromosomes, an operator used to join portions of two parents and create new offspring, which by inheritance acquire the feature of an individual and mutation (An incremental amend is prepared to each offspring chromosome of population) Operations over parent chromosome in successive generations of offspring. The offspring replaces the parent chromosome which is less fit called fitter offspring, and they become parent chromosome fro forthcoming generations. This process of replacing parent chromosomes with offspring chromosomes is called reproduction. Where the process increase the fitness average of the population. Until termination state is noticed the whole process will be iterating. Maximum iterations or maximum fitness value may be attained by termination condition. Approaches like Evolutionary Programming, Evolution Strategies are few look alike approaches to genetic algorithm which is a classification of Evolutionary computation. Based on natural selection and natural genetics, genetic algorithms are inclusive as search algorithms.

Second generation population evolution is a step further to select the genetic operands from the solution. Crossover may be called as recombination. A pair of 'parent' solutions are necessary for every new solution production, the breed for earlier selected pool. By producing a child, parent share almost every characteristics of it to every new child selected by new parent. This process propagates until appropriate sized population for a solution is generated. Couple of parents generate biologically inspired child, but more than that can generate better quality chromosomes.

III. FAULT TOLERANCE SYSTEM BASED ON GENETIC ALGORITHM

Monitoring the active unit for fault is always a continuous process. One of the best methods to find the existence of faults in an active unit is BIST architecture.

BIST being centralized active block of the whole Fault tolerant System working units will be connected to it. BIST will be continuously applied with input and output through active unit; moreover clock pulses and pulse rates are always under survey of BIST. If any fault occurs then BIST will trigger the active unit and fault to overcome.

© www.ermt.net All Rights Reserved

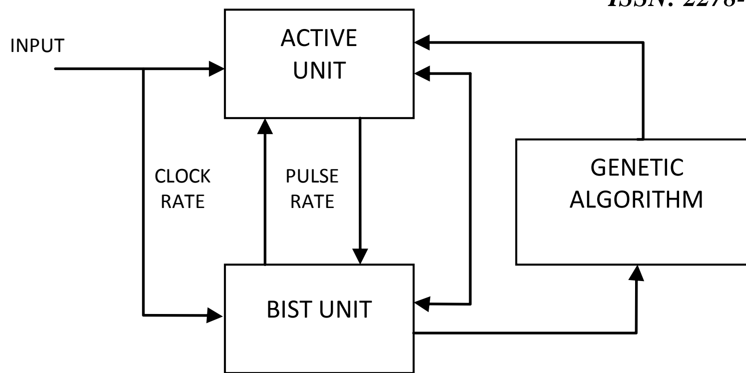


Figure 2.1 Fault Tolerance System based on genetic algorithm.

The major components of BIST architecture are

1. Test Pattern Generator
2. Test Controller
3. Output Response Analyzer

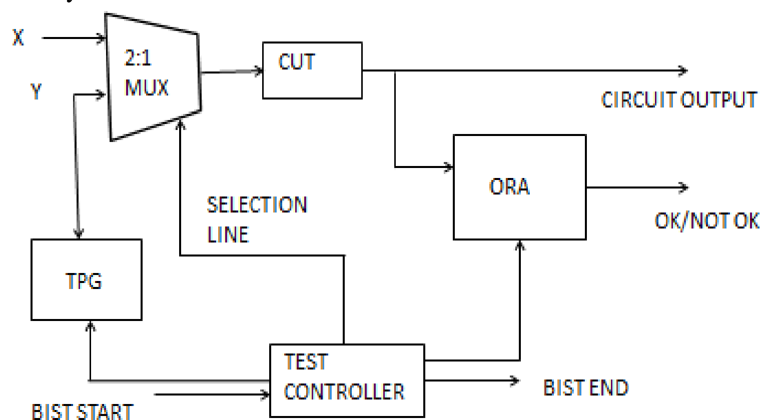


Figure 2.2 BIST architecture

3.1 Test Pattern Generator

Test vectors are a set of bit streams generated and applied them to circuit under test in a sequence or may be random sequence. Various test patterns generators practically existing are Linear Feedback Shift Register (LFSR), Counters; major contributes to TPGs by implementing in hardware

3.2 Test Controller

Circuit under test will be put in test mode by test controller and allows to drive directly the circuit inputs of test pattern generator and test controllers. Test controller is responsible for seed values which are to be abounding to TPG, based on its value. During test sequence application, Output response analyzer gets interacted with test controller and has to ensure the exact signals to compares. The controller needs to assess how many number of shift commands are desired for scan based testing. It always takes into consideration to maintain the data of number of patterns processed by the circuit. Completion of test process is indicated by the test controller after enabling output signal which states the output response analyser in determining the state such as fault free circuit.

3.3 Output Response Analyzer

This analyser compares the output generated by circuit under test and has to confirm the designed output response. Circuit can be assumed as fault free if it delivers desired response to all applied test vectors. Otherwise the circuit assumed to be faulty if delivers incorrect response for more than majority test vectors.

IV. DETECTION OF FAULT IN MEMORY

Memory chip cannot be put for online test when its being used in any system, the only reason is it would erase the contents of the memory. A caution to be aware of is memory contents. Online testing is one of the best capabilities of BIST, which has highest potential of diagnosing fault. This diagnosing method supports to reconfigure the memory unit and rectify the fault present in the module. As a matter of fact BIST logic can be implemented for testing circuits and manufacture. BIST incorporation can be best suited for memory when its power on and memory unit contains data which is in use for fault processing. A fault, so called Stuck-at-fault either stuck-at 0 or stuck-at 1 of the circuit is an assumption of fixed logic, stuck-at- fault can be detected using limited number of test vectors. The target node should be set to 0 for stuck-at 1 and the same should be changed to 1 for stuck-at 0, that affects the output from the applied propagation by the test vectors.

© www.ermt.net All Rights Reserved

4.1 Corrections of Faults

Existing systems are upholding a spare cell i.e. chip based application for permanently but the existing systems cannot be exempted. When a fault is detected by BIST, reset unit corrects the fault whatever the error might be like aging fault, burning of chip, over programming, benign faults etc. To refresh the result unit, reset unit plays a essential responsibility even when faults are over come. In digital systems self repairing methods are very much proficient method in correction of fault, but this method might be better option for identifying it. This method is very efficient but as par time utilization is concerned it's a little higher. Rigger the cell under working should be overcome for any states of fault to overcome.

V. THE REPAIR TECHNIQUE

After performing testing on circuit under test if any errors are identified, those errors either to be rectified for desired output or repair the component, so incomplete circuit which is erroneous should be corrected, by designing a repair component and operate in combination with circuit evolved. We connect a XOR gate as a repairing component for repairing technique. Since XOR of "0" to Boolean expression and any is equal to itself and vice versa, and XOR is "1" to any expression of Boolean yields inverse expression for given Boolean expression, as an example to display an illustration is considered as under Table.

Table 1.The Illustration truth table

Input	Output
000	0
001	0
010	1
011	1
000	**0**
001	**0**
010	**1**
011	**1**

To make this illustration more clear two different nodes are to be considered

1) Considering the circuit is erroneous for sole input only
2) Considering the circuit is erroneous for more than on inputs.

Suppose a solution of a candidate is assumed to be accurate output for input 010, say output is 1 but anticipated to be '0'and all other input are supposed to be accurate. Block AND and XOR gates are repairing components i.e. if output is '0' and XOR gate is also '0', since the circuit has corrected by repair component.

VI. GRACEFUL DEGRADATION

The methods include redundant hardware deployment for detection and recovery of faults. When a spare hardware block is utilized for faulty block replacement, graceful degradation occurs followed by system functionality degradation. In the process of fault detection every block is put under check with Triple Modular Redundancy (TMR) voter circuit. If any two circuits distinct to be voted true, identifying a faulty block. Based upon that result faulty block is replaced by a spare block available in the module. As an illustration taking into consideration a block diagram consisting four operational blocks and two test blocks. In the block diagram 1, Block 1 to Block 4 has analogous functionality and both test blocks also with similar functionality. These two test blocks will check the blocks 1 to 4 in a pattern, just through TMR scheme. If any error is recognized then the faulty block is replaced with test block, making the system to persist to execute its operation while degrading the capability of fault detection.

Detailed description is displayed with a sequence of images, where every block from 1 to 4 are tested after recognizing the faulty block it is replaced with test block

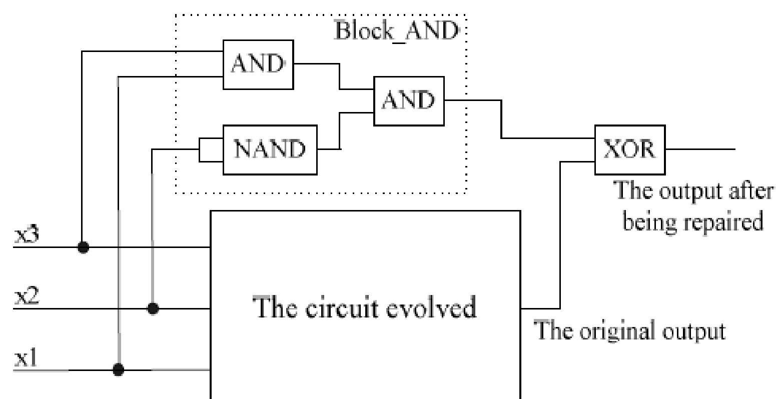


Figure 6.1. The component used for repair that is erroneous for single input

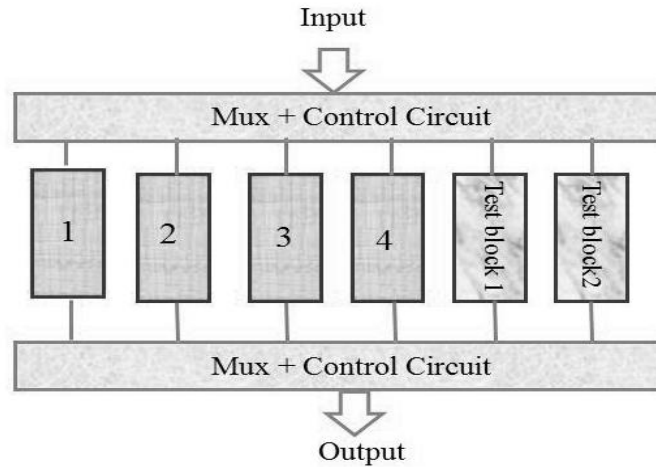
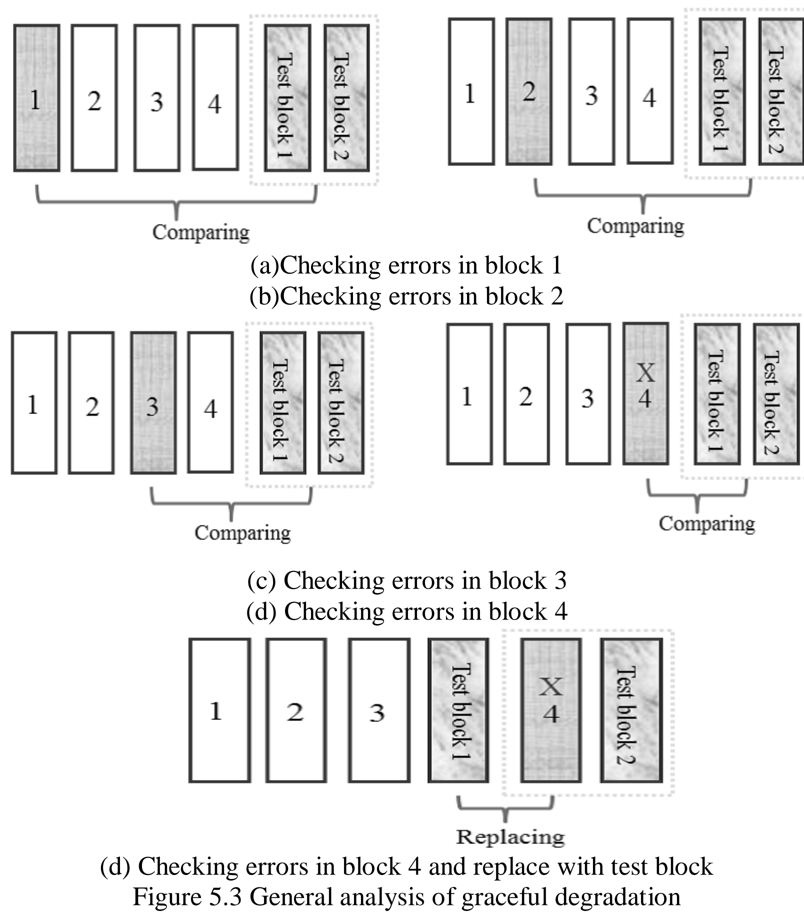


Figure 6.2. Redundant hardware deployment for detection and recovery of faults



(a)Checking errors in block 1
(b)Checking errors in block 2

(c) Checking errors in block 3
(d) Checking errors in block 4

(d) Checking errors in block 4 and replace with test block
Figure 5.3 General analysis of graceful degradation

1. Avoid soundless data error
2. Very efficiently implemented as number of gates essential with low latency.
3. Increases the fault detection capabilities at the cost.
4. The area overhead is condensed

This paper fine points the details handled by genetic to identify faults in a block applied by various patterns, this algorithm if used in automatic test pattern generation tools with dissimilar test patterns, it can detect all stuck at faults

VII. SIMULATION RESULTS

The simulation tool used for processing the input digital bits is done using VHDL. The MODELSIM is used for simulation. The version of MODELSIM is 6.5b. The input bits are get simulated to produce the output by the digital waveform.
Figure 4.1 and 4.2 shows that the Input file is assigned for the work library to develop the file which is to be created after compiling .Objects is done for the workspace that mentioned for the test bench. It has been loaded to the waveform that has to be represented.

© www.ermt.net All Rights Reserved

Figure 4.3 shows that the Wave form is generated to the objects that are created from the where the files are generated for the undefined files for xo1, xo2, xo3, xo4 etc., here the spare cells are defined for the undefined variables to be declared. Hence clock, reset and selection variable is assigned to the file.

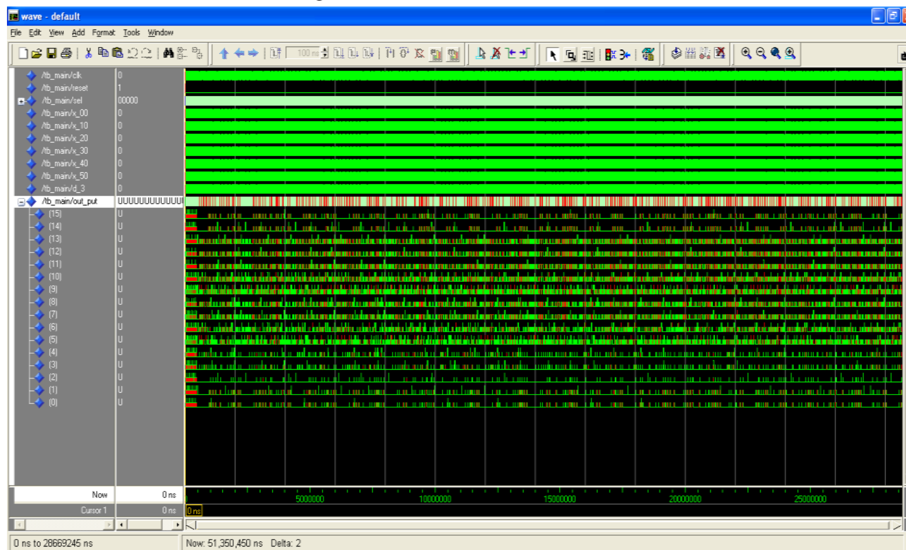


Figure 7.1 Output waveform of the working cell and spare cell

Figure 4.4 and 4.5 shows that the red color denotes the faulty cell and green color indicates the working cell. If fault occurred in the working cell it will be recover by the spare cell using genetic algorithm.

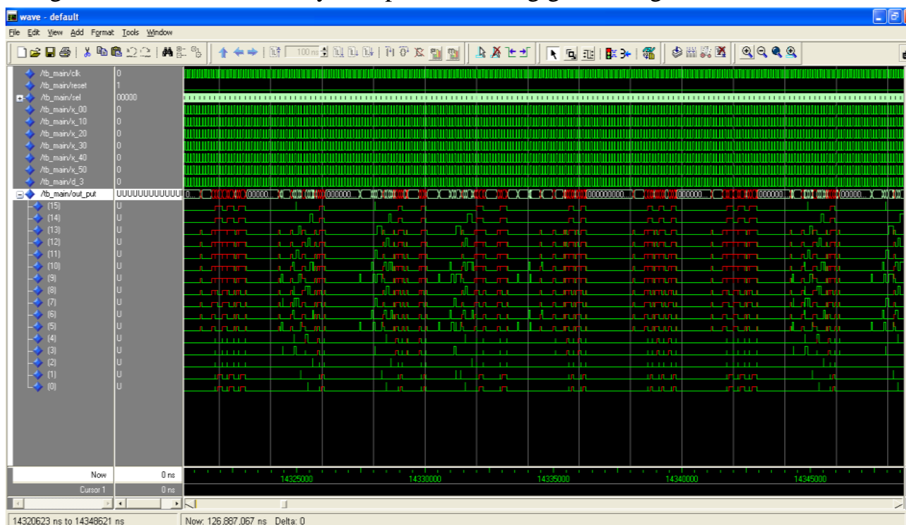


Figure 7.2 Expandable view of the cell affected areas of the particular region**.**

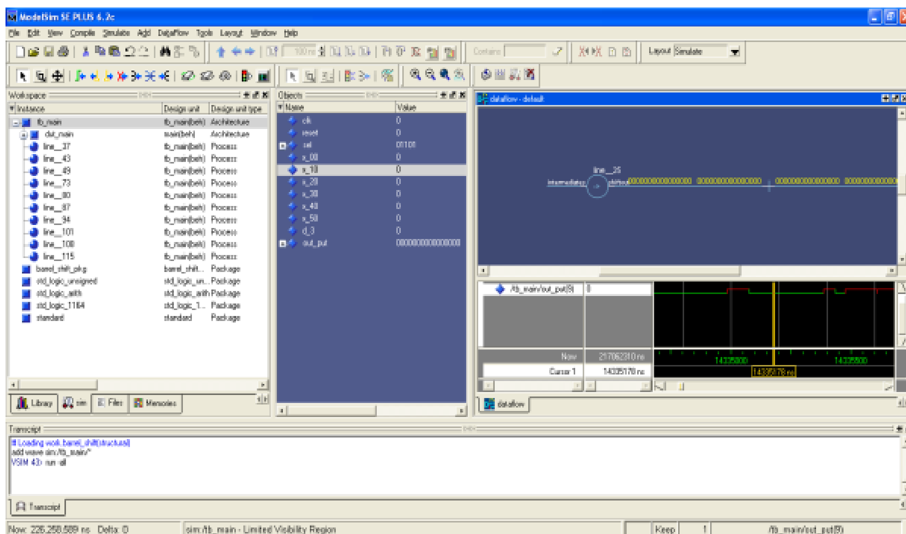


Figure 7.3 shows the clock pulse diagram

© www.ermt.net All Rights Reserved

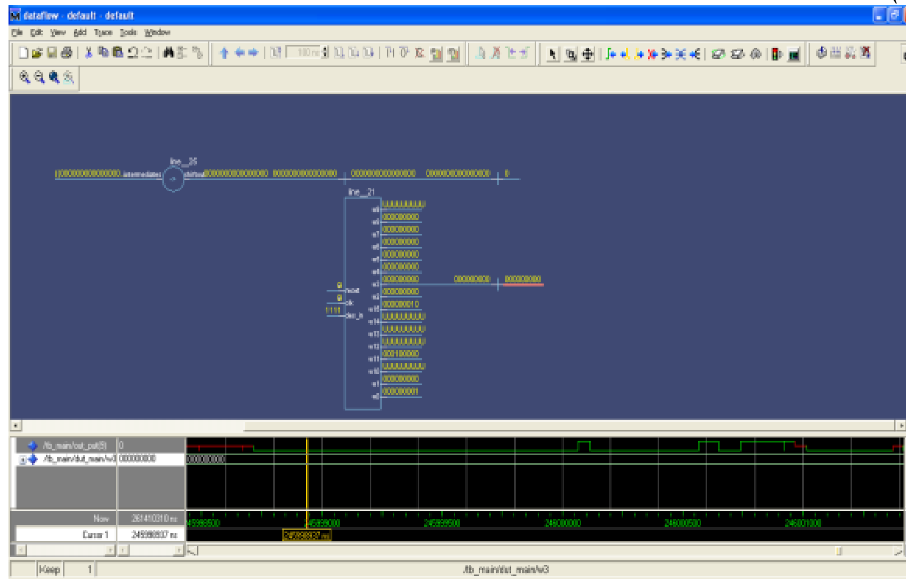


Figure 7.4 shows the input and output of the cell

Figure 4.6 and 4.7 shows the clock pulse diagram of the input and the output vector representation number of inputs and outputs to be declared in the given input and sparse cells to be represented.

VIII. CONCLUSION

Simulation results show that the arithmetic and logic unit is generated by via the genetic algorithm function healthy within the timing boundaries obligatory. Based on this basis the exploit of a genetic algorithm lead to a considerable reduction in design time while taking into contemplation the timing issues essential to today's deep sub-micron technologies. A self-repairing digital system provides high-quality scalability and fault coverage is proposed. Area overhead is reduced compared with the previous self-repairing technique. Therefore the fault tolerant or self-repairing results process is simple and fast, which facilitates the efficiency design for more secured systems.

REFERENCES

[1] M. Samie, G. Dragffy, and T. Pipe (2011) "UNITRONICS: A novel bio inspired fault tolerant cellular system," in Proc. NASA/ESA Conf. Adapt. Hardw. Syst., pp. 58–65.

[2] M. Samie, G. Dragffy, and T. Pipe (2010) "Bio-inspired self-test for evolvable fault tolerant hardware systems," in Proc. NASA/ESA Conf. Adapt. Hardw. Syst., pp. 325–332.

[3] Huan-yuTu (2010) "Comparisons of Self-Healing Fault-Tolerant Computing Schemes" Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I WCECS 2010, October 20-22, 2010, San Francisco, USA.

[4] M. Samie, G. Dragffy, A. Popescu, T. Pipe, and C. Melhuish, (2009) "Prokaryotic bio-inspired model for embryonic," in Proc. NASA/ESA Conf. Adapt. Hardw. Syst., pp. 163–170.

[5] Yocheved Dotan, Member, Nadav Levison, RoiAvidan, and David J. Lilja, Fellow (2009) "History Index of Correct Computation for Fault-Tolerant Nano-Computing" ieee transactions on very large scale integration systems.

[6] Harikishore. Kakarla, Madhavi Latha.M and Habibulla Khan (2012) "Self Correcting Memory Design For Fault Free Coding In Progressive Data Streaming Application" International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1,

[7] L. J. K. Durbeck and N. J. Macias (2002) "Defect-tolerant, fine-grained parallel testing of a cell matrix," Proc. SPIE, vol. 4867, no. 8, pp. 71–85,

[8] A. J. Greensted and A. M. Tyrrell (2005) "Implementation results for a fault tolerant multicellular architecture inspired by endocrine communication," in Proc. NASA/DoD Conf. Evolvable Hardw., pp. 253–261.

[9] P. K. Lala and B. K. Kumar (2003) "An architecture for self-healing digital systems," J. Electron. Testing: Theory Appl., vol. 19, no. 5, pp. 523– 535.

[10] D. Mange, M. Sipper, A. Stauffer, and G. Tempesti (2000) "Toward robust integrated circuits: The embryonics approach," Proc. IEEE, vol. 88, no. 4, pp. 516–541.

[11] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams (1998) "A defect tolerant computer architecture: Opportunities for nanotechnology," Science, vol. 280, pp. 1716–1721.

[12] L. J. K. Durbeck and N. J. Macias (2002) "Defect-tolerant, fine-grained parallel testing of a cell matrix," Proc. SPIE, vol. 4867, no. 8, pp. 71–85.

[13] Y. Thoma, E. Sanchez, J.-M. Moreno, J. Arostegui, and G. Tempesti (2003) "A dynamic routing algorithm for a bio-inspired reconfigurable circuit," in Lecture Notes in Computer Science. New York: Springer-Verlag, pp. 681–690.

[14] Vladimir Hahanov, Eugenia Litvinova, Wajeb Gharibi, OlesyaGuz (2009) "Algebra-Logical Repair Method for FPGA Logic Blocks" National University of Radio electronics, Ukraine.
[15] Kenneth A. Clark Alan A. Ross, Hersch H. Loomis (2003) "Modeling Single-Event Effects in a Complex Digital Device" IEEE Transactions On Nuclear Science, Vol. 50, No. 6.
[16] W. Barker, D. M. Halliday, Y. Thoma, E. Sanchez, G. Tempesti, and A. M. Tyrrell (2007) "Fault tolerance using dynamic reconfiguration on the POEtic tissue," IEEE Trans. Evol. Comput., vol. 11, no. 5, pp. 666–684.
[17] G.W.Greenwood, (2005) "On the practicality of using intrinsic reconfiguration for fault recovery," IEEE Trans. Evol. Comput., vol. 9, no. 4, pp. 398–405.
[18] D. Mange, M. Sipper, A. Stauffer, and G. Tempesti (2000) "Towards robust integrated circuits: The embryonics approach," Proc. IEEE, vol. 88, no. 4, pp. 516–541.