

# Fault Tolerant Digital System Using Self-Repairing System

M.Saranya<sup>1</sup>, S.Mahalakshmi<sup>2</sup>, V.Dhivya<sup>3</sup>

PG Student, Dept. of ECE, Syed Ammal Engineering College, Ramanathapuram, Tamilnadu, India<sup>1,2,3</sup>

**Abstract:** Reliability has always been an important issue with electronic systems ever since the first electronic systems were designed. compared to the biological systems, the electronic systems are so fragile that even a single problem can occurs then the total system useless. Therefore, devising fault-tolerant systems that can deal with such delicate problems has been a considerable challenge. During the early stages of the development of fault-tolerant systems, dual modular redundancy (DMR) and triple modular redundancy (TMR) methods were introduced. These techniques ran the same modules in parallel, and thus a faulty module could be distinguished by comparing outputs of the same modules and voting for the majority one (with TMR) or by using an additional device (with DMR). However, these methods are only detect the fault and it can't be replace the faultless part instead of faulty part in a system when the fault occurrence. The size of the module is so huge that a large part of the circuit must be replaced even if a small part in the module is malfunctioning. In this paper, we propose a self-repairing system that the faulty part or cell can be replaced by the faultless part or cell in a system..so it increases the reliability.

**Keywords:** Dynamic routing, Fault tolerant system, reliability, self-repairing, spare cell, working cell

## I. INTRODUCTION

The reliability of a system or component to perform its required functions under stated conditions for a specified period of time. It is the overall consistency of a measure. When we call someone or something reliable, we mean that they are consistent and dependable. A reliable test is one that consistently produces the same results when administered to the same individuals under the same condition. Here we achieve the fault tolerant by using self-repairing digital system. self-repairing is replaced or recognize itself when the fault occurrence. compared to this self-repairing system, redundancy has to be running all the time, and it can only cover the fault once. In the last 10 years, these conventional methods have proven to be rather inefficient, and scientists have consequently turned to biology to find inspiration for a more suitable

self-repairing circuit that can resolve the aforementioned problems and faults with fault-tolerant systems. These self-repairing circuits can also recover from a fault by isolating the faulty block and differentiating a spare block previously held by the faulty block. With such self-repairing systems make repairs on a fine-grained scale rather than the coarse-grained scale of conventional fault tolerant systems use. Because the two essential procedures of self-repair, cell replacement, and the rerouting process are highly complex, these systems are difficult to implement. The main advantage of the TMR/DMR scheme is that it can mask faults instantaneously allowing correct functioning of the circuit without interruption, since there is no need for any fault detection or recovery procedure. Hence it is used in systems employed by critical applications where even a small delay due to the occurrence of a fault can jeopardize the entire operation. However, the scheme is the rigid and expansive way of achieving fault tolerance. The power consumption is considerable since all the redundancy modules need to be powered. Further, the voter can cause system failure. To overcome this problem, a redundant voter scheme is sometimes adopted, in which the voter is also replicated. Many variations to the TMR/DMR scheme have been suggested to suit specific fault-tolerant requirements. one such schemes makes use of a threshold voter in place of the majority voter. The voter output is 1 only if the weighted sum of its input is equal to or greater than its threshold M.

## II. OVERVIEW OF THE PROPOSED SELF-REPAIRING DIGITAL SYSTEM

Self-repairing is a phrase applied to the process of recovery (generally from psychological disturbances, trauma, etc.), motivated by and directed by the patient, guided often only by instinct. Such a process encounters mixed fortunes due to its amateur nature, although self-motivation is a major asset. The value of self-repairing lies in its ability to be tailored to the unique experience and requirements of the individual. The process can be

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

helped and accelerated with introspection techniques such as meditation. NASA has successfully demonstrated an aircraft flight control feature that has the potential of saving lives and lowering aircraft maintenance costs. It is the Self-Repairing Flight Control System (SRFCS), a software addition to an aircraft's digital flight control system that detects failures and damage to ailerons, rudders, elevators, and flaps. The system which can be used on nearly all aircraft with digital flight control systems then compensates for the component loss by reconfiguring the remaining control surfaces so flight crews can land their aircraft safely. Installed on military aircraft, the unique system would allow aircrews experiencing a control surface failure to complete important tactical missions. Self repairing may refer to automatic, homeostatic processes of the body that are controlled by physiological mechanisms inherent in the organism.

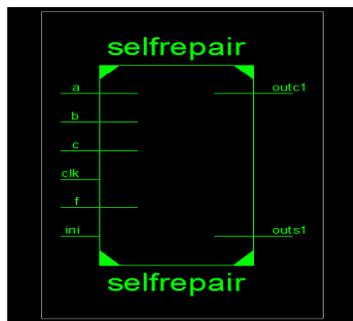


Fig 3.1: Self-Repairing system

In this paper, the proposed system simplifies the self repairing mechanism and helps the circuit to maximize the efficiency of its hardware realization. The size of the hardware can grow in proportion to the number of functional modules while also ensuring good fault-coverage. Here we fixed the four spare parts/cells(SP) around one working part/cell(WP). when the fault occurs in the working part, then it can be replaced by any of the four spare parts. we give the name for each spare parts and working part in 3 bits and 5 bits respectively.

|               |               |               |               |
|---------------|---------------|---------------|---------------|
| SP(100)<br>\  | wp<br>(00110) | SP(101)<br>\  | wp<br>(00111) |
| wp<br>(00101) | SP(000)<br>\  | wp<br>(00100) | SP(110)<br>\  |
| SP(011)<br>\  | wp<br>(00000) | SP(001)<br>\  | wp<br>(00011) |
| wp<br>(00001) | SP(010)<br>\  | wp<br>(00010) | SP(111)<br>\  |

Fig1: spare part(sp) and working part(wp) positions

In a figurative sense, self-repairing properties can be ascribed to systems or processes, which by nature or design tend to correct any disturbances brought into them. In order to separate the roles of hardware structure and control. The working part (00000) have the four spare parts. The top spare part, right spare part, bottom spare part and the left spare part names are 000,001,010,011 respectively. Like that given the name for each spare parts and working part. The positions of spare parts and working part are shown in fig1. In spite of the multiple advantages of self-repairing systems, several problems remain as major obstacles for practical use. Because the two essential procedures of self-repair, cell replacement and the rerouting process are highly complex, these systems are difficult to implement. Both the MUXTREE and Lala self-repairing systems present the basic methods for arranging modules during expansion, but these methods do not provide a complete solution that can resolve the problem with good fault-coverage

### III. NEW SELF-REPAIRING MECHANISM

In information technology, self-healing describes any device or system that has the ability to perceive that it is not operating correctly and, without human intervention, make the necessary adjustments to restore itself to normal operation. Because users of a product may find the cost of servicing it too expensive (in some cases, far more than the cost of the product itself), some product developers are trying to build products that fix themselves. IBM, for example, is working on an [autonomic computing](#) initiative that the company defines as providing products that are self-configuring, self-optimizing, and self-protecting - as well as self-healing. For all of these characteristics together, IBM uses the term "self-managing."The system

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

must assign the proper module to replace a faulty one, and the substitute module must be connected to neighbouring modules in the same way that the faulty module was previously connected.

different concepts in Windows Installer which people many times consider to be the same thing, however there is difference in these two.

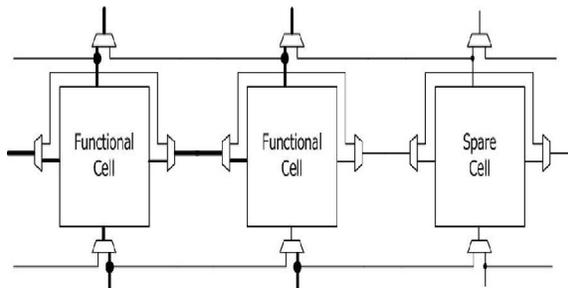


Fig. 2: state before the fault occurrence

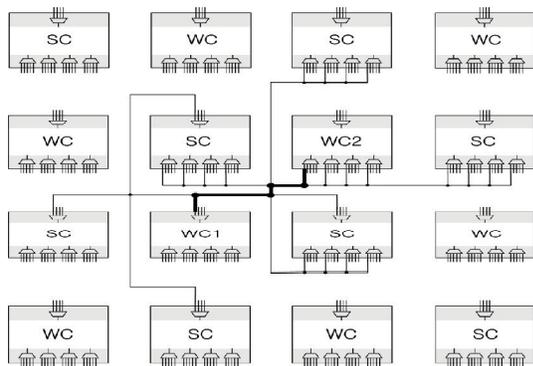


Fig 3: wc1 and wc2 working normally before faultoccurring

Therefore, such methods of self-repair involve both additional hardware for rerouting after the replacement of faulty cell (module) and inefficient arrangement of functioning modules as well as spare (stem) modules. As the circuit size increases, the size of the spare modules and additional modules beside the functional modules exponentially increases. Furthermore, if there is no available spare module, existing self-repairing circuits must dispose off the entire group of modules, even if some of them are still functioning. Before the fault occurrence of a system, the two functional cells are working normally and connected one spare cell for the replacement after the fault occurrence shown in fig2 and it is explained in detailed in fig3. Here we assume that the name for working part or cell 00000 is wc1 and 00100 is wc2. Here the both working cells are working normally before the fault occurrence. Self Heal and Repair are two

Self Heal is triggered by advertised shortcuts, or other advertising information in the package which eventually Repairs the application. When the application is launched by advertised shortcut, it checks for all the key paths of the Current Feature, if any of the key paths is missing it will launch Repair. To realize our routing architecture, the WCs and SCs are arranged as shown in Fig. 3. In Fig. 3, if the output of WC1 is connected to the input of WC2 while two WCs can be replaced by neighbouring SCs, two types of routing must be made above all. WC1, which sends the output, is connected to four neighbouring SCs while WC2, which receives the output of WC1 as its input, is connected in the same way (the dotted line in Fig. 3). For each type of line in Fig. 3, the same order of inputs of MUXs among cells must be connected. Then, two types of wires are connected. The distant connection block which is not illustrated in Fig. 3, has the same types of connections. As shown in Fig. 3, the proposed routing architecture is composed of connection wires and input selection MUXs in order to dynamically connect the output of WC1 (or the SCs of WC1) to the input of WC2 (or the SCs of WC2). The real connection among inputs and the outputs of the cells are controlled by the genome of each cell. For example, if the output of WC1 is initially connected to the input of WC2 through the first MUX in WC2, all of the other inputs and outputs of cells that are connected to the wire are disconnected. The outputs of SCs are blocked such that they do not interfere with other operating cells. The outputs of spare cells and faulty cells are isolated by a tri-state buffer. As a result, the substituting SC of WC2 can receive the output of WC1. The priority of spare parts for differentiation is given in a counter clockwise direction from the left SC of the WC. Therefore, the algorithm sets the system to skip the other working or isolated cell and differentiates the next spare cell which has not yet been used. If there are no more spare cells for fault recovery, the system stops operating and moves on to system failure.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

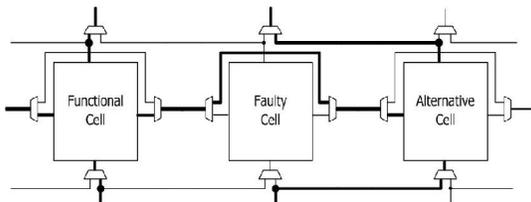


Fig 4: state after the fault occurrence

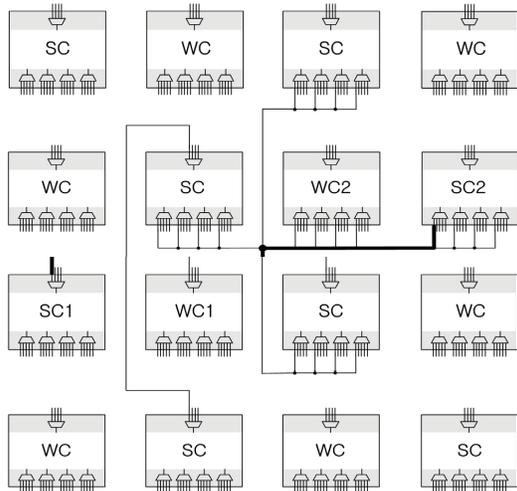


Fig 5: spare cell is replaced by faulty cell

Hence, the proposed routing architecture has an efficient structure in which routing changes in a complex circuit can easily be done despite the small number of wires in the circuit. suppose the fault occurs in the system then it can be replaced by the spare cell connected in the system for the replacement. Assume that the second functional cell is going to faulty then it can be replaced by the spare cell shown in fig4 and it is explained in detailed in fig5. Here we assume that the name for spare part or cell 011 is sc1 and 110 is sc2. Here the working cells wc1 and wc2 are replaced by the spare cells sc1 and sc2 respectively.

#### IV. EXPERIMENTAL RESULTS

We inject a permanent fault when the fault is detected again at the rising of the next clock after the faulty data are recovered to normal data, otherwise, it is indicating a transient fault, in which case the system operates normally. First, in order to verify the system, four cases of

recovery from a permanent fault are demonstrated. Similarly, SP(000) shows the changed states of cells by the fault recovery after a fault is sequentially injected to S2 and S3, respectively. Here, the system repairs the sequential permanent fault in a counterclockwise sequence. However, if the fault is generated in S4 after a faulty S3 is replaced by S4, the entire system stops operating because there is no SP left for repair. fault recovery by skipping the used SP is demonstrated. If S3 is already occupied and a fault in S4 is generated, S3 is skipped and S4 is replaced by S5.

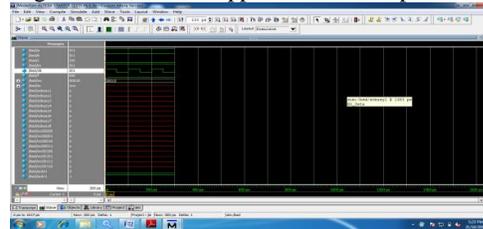


Fig 5.1: Simulation output when spare cells are not busy

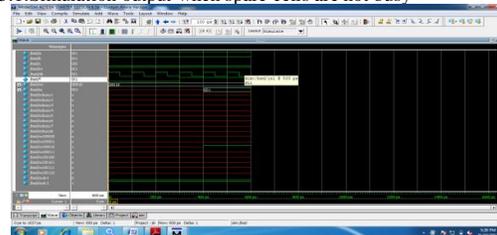


Fig 5.2: Simulation result when the spare cell(000) is busy

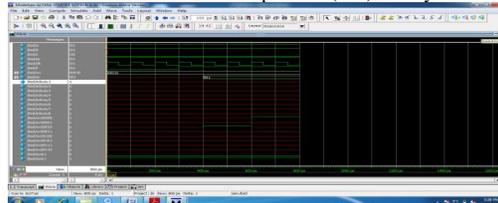


Fig 5.3: Simulation result when the spare cell(001) is busy

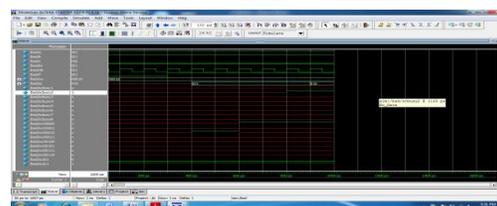


Fig 5.4: simulation result when the spare cell(010) is busy

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3 , Issue 4 , April 2014

The recovery from two simultaneous permanent faults is also demonstrated. If faults in W1 and W2 are generated simultaneously, they are replaced by S1 and S4, respectively. Secondly, the transient fault is demonstrated according to SP(001). When the spare cells are not busy then it takes any of that spare cells for replacement instead of faulty cell. Suppose, the spare part(000) is busy then it chose the spare part (001), that cell is also busy then it takes the next spare part (010), that cell is also busy then it replaced by the next spare part (111). So there are four spare cells means that four replacement around one working cell for replacement.

## V. CONCLUSION

In this paper, a new self-repairing digital system providing good scalability and fault coverage was proposed. New architectures for the routing and the cell(part) were developed and well organized such that rerouting between neighboring cells after the replacement of a faulty cell would be done by only replacing the faulty cell with an SP. Furthermore, due to the new architecture, the cells(parts) could be arranged in a flexible manner such that the WP could be expanded to any four directions, and could also be arranged densely such that SP could be replaced by any of four neighbouring WPs for fault recovery without collision due to exact control. As a result, all these make the system efficient. The proposed system was compared with other major self-repair approaches and it was found that the proposed system has good fault coverage, low overhead, and no unutilized resources for fault recovery. For purposes of this paper, the term reconfigurable flight control is used to refer to software algorithms designed specifically to compensate for failures or damage of flight control effectors or lifting surfaces by using the remaining effectors to generate compensating forces and moments. This paper will discuss influences on the development of the concept of control reconfiguration and initial research and flight-testing of approaches based on explicit fault detection, isolation, and estimation as well as later approaches based on continuously adaptive and intelligent control algorithms. Also, approaches for trajectory reshaping of an impaired aircraft with reconfigurable inner loop control laws will be briefly discussed. Finally, there will be some discussion of current implementations of reconfigurable control to improve safety on production and flight test aircraft and remaining challenges to enable broader use of the technology such as the difficulties of flight certification of these types of approaches. It can be

used in satellite and also used in medical field. For further improvement of the proposed self-repairing system, there remain several issues awaiting further studies. The function of the target system is represented in the proposed system framework and this target system is to be operated as fault-tolerant or self repairing. In the proposed system framework, we did not intend to implement such extra hardware for the possible secondary fault in the additional functional hardware, but rather focused on the faults that can occur in the application circuit. The proposed work is the four spare cells are reduced to two spare cells for minimizing the area. Because the four replacement is unnecessary. On the other words, here the four spare cells are there, that much of faults are not possible in any of the electronic systems.

## REFERENCES

- [1] W. C. Carter, "Fault-tolerant computing: An introduction and a view-point," *IEEE Trans. Comput.*, vol. 22, no. 3, pp. 225–229, Mar. 1973.
- [2] C. Ortega and A. Tyrrell, "Design of a basic cell to construct embryonic arrays," *IEE Proc. Comput. Digital Tech.*, vol. 145, no. 3, pp. 242–248, May 1998.
- [3] X. Zhang, G. Dragffy, A. G. Pipe, N. Gunton, and Q. M. Zhu, "A reconfigurable self-healing embryonic cell architecture," in *Proc. ERSA*, Jun. 2003, pp. 134–140.
- [4] D. Mange, E. Sanchez, A. Stauffer, G. Tempesti, P. Marchal, and C. Piguet, "Embryonics: A new methodology for designing field programmable gate arrays with self-repair and self-replicating properties," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 6, no. 3, pp. 387–399, Sep. 1998.
- [5] D. Mange, S. Durand, E. Sanckez, A. Stauffer, G. Tempesti, P. Marchal, and C. Piguet, "A new paradigm for developing digital systems based on a multi-cellular organization," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, Apr.–May 1995, pp. 2193–2196.
- [6] P. K. Lala and B. K. Kumar, "An architecture for self-healing digital systems," *J. Electron. Testing: Theory Appl.*, vol. 19, no. 5, pp. 523–535, Oct. 2003.
- [7] N. J. Macias and L. J. K. Durbeck, "Adaptive methods for growing electronic circuits on an imperfect synthetic matrix," *Biosystems*, vol. 73, no. 3, pp. 173–204, Mar. 2004.
- [8] Y. Thoma, E. Sanchez, J.-M. Moreno, J. Arostegui, and G. Tempesti, "A dynamic routing algorithm for a bio-inspired reconfigurable circuit," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2003, pp. 681–690.
- [9] A. J. Greensted and A. M. Tyrrell, "An endocrinologic-inspired hardware implementation of a multicellular system," in *Proc. NASA/DoD Conf. Evolvable Hardw.*, 2004, pp. 245–252.
- [10] A. J. Greensted and A. M. Tyrrell, "Implementation results for a fault-tolerant multicellular architecture inspired by endocrine communication," in *Proc. NASA/DoD Conf. Evolvable Hardw.*, 2005, pp. 253–261.