Middle-East Journal of Scientific Research 18 (10): 1385-1389, 2013
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2013.18.10.12401

Hardware Reusable Fpga Design

[1]P. Arun Kumar, [2]P. Pandian and [3]Raja Paul Perinbam

[1]SENSE School, VIT University, Vellore-632014, India
[2]SAS, VIT University, Vellore-632014, India
[3]E.C.E Department, KVCE, Chennai, India

Abstract: We discuss a fault tolerant technique for Field Programmable Gate Arrays (FPGAs) called as hardware reusable when working in real time environment when the system is subjected to external attacks namely transient attacks. The proposed design, hardware reusable architecture makes its own components to overcome the faulty ones thus never making use of spare and it dynamically reconfigures the faulty component by sending the test signals from where the data got lost. The reliability of the system is measured by the attempt to shoot up at a short span of time to start its process. In this attempt, we can make 100% Fault Coverage (FC) when compared to the previous techniques of Total Modular Redundancy.

Key words: FPGAs · Fault Tolerant · Fault Detection · Routing · Logic Elements

INTRODUCTION

Today in real word applications, hardware and software co-simulation have become an integral part in complex processors. This system exists in the environment and provides data exchange and executes for a period of time called as Run Time. This general purpose computer has a set of hardware components acts as hardware accelerators to speed up the system. With the latest trends in technology, many complex processors and System On Chips (SOCs) have been fabricated into FPGAs which enables the system to work even if high computation is required. The main advantage of FPGA is reconfiguration where the user can modify the program whenever necessary depending upon the application which makes the programmer more flexible to develop his own design and dump it into the hardware. Another important advantage is that they contain pure logic elements which make it more suitable for high end applications. Now-a-days, all FPGAs employ SRAM based memory device which is useful for longtime applications as reconfigurability is the major concept which makes the design high dependability for wide range of applications. The Combinational Logic Blocks (CLBs) in FPGAs which contains both combinational and sequential circuits are programmed by the bit stream by loading the configuration data which frequently modifies the previously stored design thus making the processor more reliable.

The execution time of FPGA is very small when compared to Real Time Operating Systems (RTOS). FPGA implements all logic functions which are dumped into the hardware by using the bit streams by using JTAG cable. The device can program any application without any restrictions and enables us to reconfigure the hardware which includes combinational, sequential circuits and memory devices. This principle mainly differentiates FPGA from Application Specific Integrated Circuits (ASICs) which are dedicated to a specific application and in order to change the function, the user has to start from the scratch. As FPGAs are intended for high end applications where the frequency varies from MHz to GHz, synchronization has to be made between the two communicating devices.

Once the frequency synchronization is met, a handshake protocol occurs between the two devices which are used in real world application for the latest technology devices. Other factors to be taken into consideration of complex designs are selection of logic devices which makes the application for faster performance without any barriers and consideration of clocks which will reduce the skew rate.

Corresponding Author: P. Arun Kumar, SENSE School, VIT University, Vellore-632014, India.

Previous Work Done in Fault Tolerant Schemes: Reliability and Fault Tolerance are the main concepts to improve the performance of the system. To ensure the quality of the system, the stability of the system to shoot up after the fault occurrence is an important factor to determine the reliability of the system. As FPGAs are used for complex high end applications, they are often subjected to transient faults called as scrubbing. This is called as Single Event Upset (SEU) which will alter the data stored in memory cells. To overcome this effect, dynamic partial reconfiguration has been designed [1] which replaces the defective component by a spare but the disadvantage is it requires more slices to replace the defective module and also the spare units are to be mapped to the reconfigurable areas which increases the area overhead and degrades the system performance.

To prevent uncontrolled shut down of the system in real time applications, back to back converters were developed [2] which acts as a bidirectional switch to isolate the fault. This type of application is mainly used in power generation units as shown in fig. 1

Many techniques utilize multiplexers and D-FF for the diagnosis of faults as shown in Fig. 2. In this method, the Look Up Table (LUT) [3] is configured by using Walsh Codes and the outputs are configured by using Shift Registers which monitors the actual and the expected outputs at every time interval of 1µs.

The test vectors are generated by Automatic Test Equipment (ATE) and the script files which generate the NCD file which describes the FPGA test circuit and the faulty LUT is replaced by the spare LUT. The disadvantage of this method is the generation of NCD file.

Reconfigurable systems are mainly affected by Soft Errors which will corrupt the data and leads to drastic changes in functionality and performance of the system. In order to overcome this problem, coarse grained reconfigurable architectures have been proposed which has the advantage of reduction in configuration time and reduction in placement and routing which are mainly done in operator level [4] which are mainly implemented by self checking circuits but it increases the area overhead of the design being implemented.In Triple Modular Redundancy (TMR), the outputs of each device are connected to voter logic as shown in fig. 3. In this method, the faulty component is replaced by the other working designs [5]. In this case, every memory is connected to the voter logic which monitors the output of the device. If memory 1 becomes failure, the voter generates an error signal and it is replaced by the other two working devices.

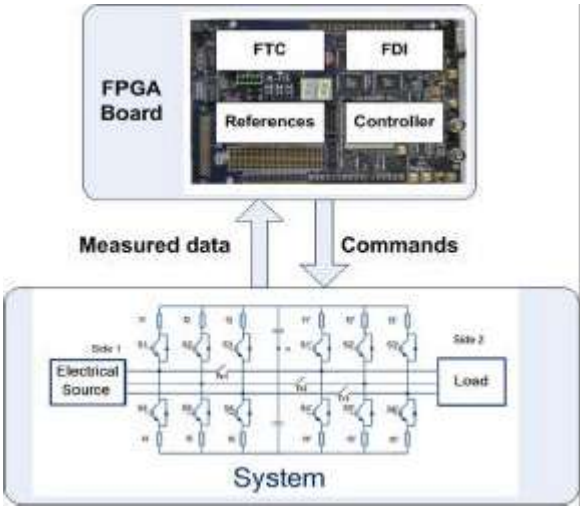

Fig. 1: FPGA based Power Generator

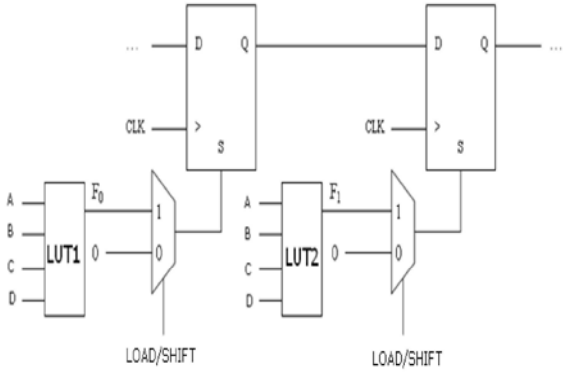

Fig. 2: LUT based Testing

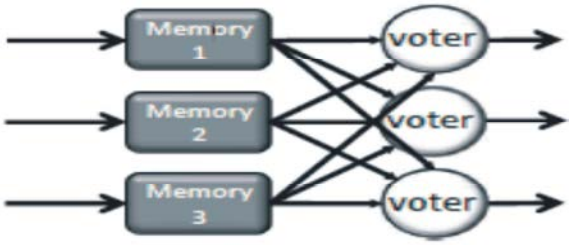

Fig. 3: TMR Design

The main disadvantage of this system is that the area and power of the design increases rapidly even though it provides 100% efficient fault tolerance.

Many real time application systems utilize Partial Reconfiguration (PR) techniques which are based on duplex architectures with checkers [6] as shown in fig. 4. It mainly works on the principle of self checking designs. The disadvantage of this system is the time taken to reconfigure the system is 56µs.

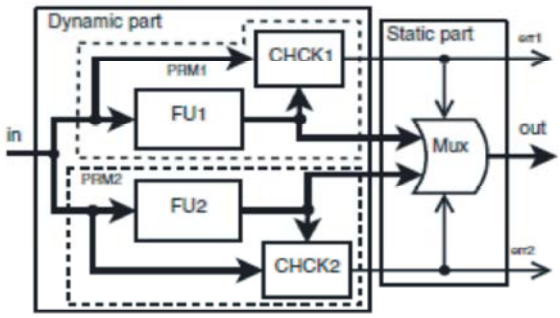

Fig. 4: Duplex Architecture with Checkers

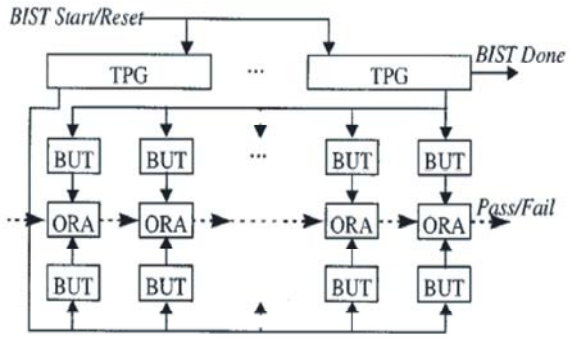

Fig. 5: Pseudo Random Testing

The bus architecture as used in FPGAs which connects any Functional Units (FUs) to the memory is very sensitive to errors. In order to protect it, the controllers of the memory devices are equipped with checking unit which mainly protects the soft errors [7] but the area overhead increases with the increasing frequency.

The Built in Self Test (BIST) based approach mainly contains Test Pattern Generators (TPGs) and Output Response Analyzers (ORA) as shown in fig. 5. In this method the Test Vectors are passed into the Blocks under Test (BUT) [8] and their outputs are measured by ORA. This mainly utilizes the principle of pseudorandom testing.

The main advantage of this method is the number of configurations used for testing and diagnosis but the drawback is fault location.

Apart from BUT, complete testing has to be provided for Programmable Logic Blocks (PLBs) which is the heart of the program circuitry of the FPGA. The technique proposed is Optimized Reconfigurable Cell Array (ORCA) which does not require storage vectors [9] and it provides maximal fault coverage of 98.2% further reducing diagnostic run time.

The basic objective of all the designs mentioned above is to identify the fault and rectify it within a short span of time which will increase the life time of the system. The disadvantage of these systems is it increases the hardware used for testing. These techniques protect the system from soft errors by replacing the faulty component by the spare components which in terms increase the area overhead and automatically degrades the system performance which are intended for long time applications used in real world even though high efficiency of fault tolerance is utilized where high computation is required to eradicate the fault.

Method Proposed for Fault Tolerant Approach: Considering the previous approaches[1-9], which relies more on the hardware resources i.e. the area overhead increases which the inclusion of BIST, we employ a new technique called as hardware reusable during the faulty conditions. In this technique, the designed hardware itself acts as a duplicate hardware to the faulty ones and thus, there is no use of going to the spare components thus saving the area overhead. This is a new technique in interconnect modeling where similar hardware whose functionality are same are interconnected together. The basic advantage of this technique is that it never uses spare components and also partial reconfiguration techniques. The fault coverage utilized in this technique is 100% and within a short span of time, the faulty hardware is replaced. Consider the block diagram as shown in fig. 4. where each Functional Units which are dynamic in nature are passed through checkers to detect any fault and the output is transferred to the multiplexer whose output is always the non faulty one. The modified version is shown in fig. 6. where the objective of the checkers is to monitor the outputs of the functional units and report the error. The error signal activates the next hardware which is being interconnected to the Faulty one.

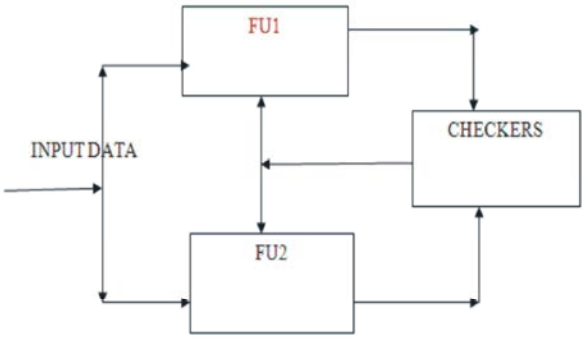

Fig. 6: Hardware Reusable Concept

The main objective to be taken in this interconnection is that the Functional Units should have the same operation so that the system will shoot up in case of failure.

Apart from that every block has a dummy register where all the results are stored in it dynamically. The checker verifies the output of the Functional Units with the dummy registers data and also reports the error. This error signal sends the activation to the next hardware which is being interconnected whose functionality is being similar to the faulty one.

The faults are injected into the system by transient methods when subjected to the run time mode in real time application systems. During the normal operation, the computations of the functional units are stored in temporary registers. Once the fault occurs in the system i.e. in FU1, the temporary register values are transferred to FU2 and make the system to work in normal operation. Thus FU2 acts as a spare for FU1 i.e. it acts as a duplicate hardware for the faulty one. This type of system utilizes fully reconfiguration technique and the area overhead is also reduced as the design is made by taking into the considerations of area, power and delay. The interconnection is made to the nearest component (similar functionality) in order to reduce the delay of the system during fault conditions.

The test patterns are generated externally by using the Automated Test Equipment (ATE), the design has been made by pipelined application and also provides pseudo exhaustive testing and also flexible which means it can be reprogrammed whenever necessary and also makes the designer aware of the fault so that he can rectify it otherwise which will lead to system failure. The backup provided to the inferior components are given by the internal hardware itself so that the area gets reduced tremendously. This type of architecture allows easy visualization of the design for the identification of the fault by using automated fault detection system and to rectify it so that the faulty component gets separated from the working component. This type of mature system helps us to find the fault within a short span of time and also becomes beneficial in many aspects of correcting it. It also provides feasibility in completing the test for all non trivial systems which are connected in parallel. It also provides extensive validation of all blocks which are to be modeled for fault simulation.

RESULTS AND CONCLUSION

From the above results we conclude that even though the methods provide 100% Fault Coverage (FC), their area and reconfiguration time seems to be more for their technique adopted. Our method provides less hardware and reconfiguration times as shown in Table 1 and 2 and also 100% FC thus satisfying the BIST concept. The future development would be increasing the fault concept to two or more hardware and employing the reusable technique.

Table 1: Area Occupied

Methodology Proposed	Area Occupied (Slices) / μm^2	Area Occupied for the Proposed technology (Slices) / μm^2
[1]	528	425
[3]	4656	
[4]	118050 μm^2	106547 μm^2
[5]	568	425
[6]	320	
[7]	580	
[8]	875	
[9]	2224	

Table 2: Reconfiguration Time

Methodology Proposed	Reconfiguration Time (µs)
[1]	60.87
[3]	75.21
Proposed Technique	30.76

Table 3: Fault Coverage

Methodology Proposed	Fault Coverage %
[1]	100%
[3]	97%
[4]	98%
[5]	99%
[6]	97%
[7]	98%
[8]	99%
[9]	98.2%
Proposed	100%

REFERENCES

1. Mihalis Psarakis and Andreas Apostolakis, 2012. Fault Tolerant FPGA Processor Based on Runtime Reconfigurable Modules, 2012 European Test Symp. (ETS), on IEEE.
2. Mahmoud Shahbazi, Philippe Poure, Shahrokh Saadate and Mohammad Reza Zolghadri, 2013. FPGA-Based Reconfigurable Control for Fault-Tolerant Back-to-Back Converter Without Redundancy, IEEE Transactions on, 60(8): 3360-3371.

3. Nandha Kumar, T., 2010. Fault Propagation and Diagnosis of LUTs in an FPGA without Fault Free Assumptions, 2010 International Conference on Electronic Devices, Systems and Applications (ICEDSA), IEEE.
4. Syed, M.A.H. Jafri, Stanisław J. Piestrak, Olivier Sentieys and Sebastien Pillement, 2010. Design of a Fault-Tolerant Coarse Grained Reconfigurable Architecture: A Case Study, 2010 11th International Symposium on Quality Electron Design, on IEEE.
5. Nathaniel Rollins, Megan Fuller and Michael J. Wirthlin, 2010. A Comparison of Fault-Tolerant Memories in SRAM-Based FPGAs, 2010 International Conference on Aerospace, on IEEE.
6. Martin Straka, Jan Kastil and Zdenek Kotasek, 2010. Modern Fault Tolerant Architectures Based on Partial Dynamic Reconfiguration in FPGAs, International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), on IEEE.
7. Martin Straka, Jan Kastil, Jaroslav Novotny and Zdenek Kotasek, 2011. Advanced Fault Tolerant Bus for Multicore System Implemented in FPGA, 2011 International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), on IEEE.
8. Miron Abramovici and Charles E. Stroud, 2001. BIST-Based Test and Diagnosis of FPGA Logic Blocks, IEEE Transactions on, 9(1): 159-172.
9. Charles Stroud, Srinivasa Konala, Ping Chen and Miron Abramovici, 1996. Built-In §elf-Test of Logic Blocks in FPGAs (Finally, A Free Lunch: BIST without Overhead!), 14th VLSI Test Symposium on IEEE.