ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

Fault Tolerance Technique for Dynamically Reconfigurable Processor

[1]Julia Mathew, [2]R.Dhayabarani

[1.] P.G Scholar, Department of Electronics and communication Engineering, V.S.B Engineering College, Karur-639111, Tamilnadu, India

[2.] Assistant Professor, Department of Electronics and Communication Engineering, V.S.B Engineering College, Karur-639111, Tamilnadu, India

ABSTRACT: This paper proposes a new technique to detect and eliminates temporary faults on FPGA systems. Soft core processors which can alleviate radiation induced failures is implemented on Virtex-5 FPGA's. This Fault tolerant technique is implemented using TMR .It recovers from configuration upsets through partial reconfiguration combined with roll-forward recovery .The lockstep scheme used here eliminates configuration upsets without interrupting normal functioning. Main Significance includes less time overhead and reduced hardware usage. Fault injection Experiments are used for the validation process.

KEYWORDS: Roll forward error recovery, Fault Tolerance, Partial Reconfiguration,Lockstep scheme,Triple modular redundancy.

I.INTRODUCTION

FPGA's are attractive in mission critical applications. Xilinx FPGA includes two types of processors: the hardcore processor and soft core processor. Hardcore processors are hardwired on FPGA die and their number is limited. Whereas soft core processors are reconfigurable and their number depends on the device size only. SRAM based technologies are very much affected by noise and produces soft errors. The soft errors changes logic states of the memory elements not inducing any permanent errors. The error caused bits can be classified either as Sensitive or as Non-Sensitive .Sensitive bits are the one's which is actually used by the design. These bits either causes Persistent Errors or Non-Persistent Errors.Non-Persistent errors disappears once the device is Reconfigured. Since it affects combinational circurity of the design .Whereas Persistent Errors affecting sequential part of the design and it is not disappearing even after the device is Reconfigured then full module based Reconfiguration and internal Reset is required .Fault tolerance technique is used to ensure reliability of FPGA systems. These techniques are used for fast fault allocation and permanent fault repair through Partial reconfiguration. Here the design is implemented using Virtex- 5 FPGA using a Lockstep scheme built using a pair of Micro blaze Core's.Inorder to identify faulty core a specially designed Pico Blaze Core is used. Once the error is detected using Lockstep Scheme and Pico Blaze core it is corrected using Partial Reconfiguration combined with Roll Forward Error Recovery Technique.

II.RELATED WORKS

Fault Tolerant Soft core processors which uses TMR is exisisting which corrects error using Roll Forward Error Recovery but it consumes 200 percent extra hardware resources which limits the possibility of building more powerful system. A Fault Tolerant version of open source LEON-3 FT was proposed previously .it allows detection and correction of errors caused by SEU's in processing unit or memory however it consumes more Resources.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

III.FAULT TOLERENCE

It is the property that enables a system to continue its operation properly in case of failure of one or more components. Its applicable in life critical systems. Fault Tolerant design can provide dramatic improvements in system availability and lead to a substantial reduction in maintenance costs as a consequence of fewer system failures.

Recovery from errors in fault-tolerant systems can be characterised as either **roll-forward** or **roll-back**. When the system detects that it has made an error, roll-forward recovery takes the system state at that time and corrects it, to be able to move forward. Roll-back recovery reverts the system state back to some earlier, correct version, for example using check pointing, and moves forward from there. Roll-back recovery requires that the operations between the checkpoint and the detected erroneous state can be made idempotent. Some systems make use of both roll-forward and roll-back recovery for different errors or different parts of one error.

IV. PARTIAL RECONFIGURATION OF FPGA

Partial Reconfiguration is the ability to dynamically modify blocks of logic by downloading partial bit files while the remaining logic continues to operate without interruption. Xilinx Partial Reconfiguration technology allows to change functionality on the fly, eliminating the need to fully reconfigure and re-establish links, dramatically enhancing the flexibility that FPGAs offer. The use of Partial Reconfiguration can allow designers to move to fewer or smaller devices, reduce power, and improve system upgradability. Make more efficient use of the silicon by only loading in functionality that is needed at any point in time.

Currently, the main interest in the granularity of the FPGA programming data is related to the dynamic reconfiguration property provided by some recent FPGAs to perform online programming (dynamic reconfiguration) of a portion of their logic (partial reconfiguration) without affecting the rest of the system. The behaviour of the FPGA is determined by configuration bit streams that consist of a sequence of instructions and control signals data. Downloading this sequence allows to program the FPGA to perform requested design functions.

A nonreciprocal relation exists between the design and its bit stream: it is not possible to extract the design structure and implementation on the FPGA from the bit stream. The reconfiguration process itself can be done either completely or partially by sending the related bit stream (full or partial) to the Internal Configuration Access Port (ICAP).

V.FAULT TOLERANT ARCHITECTURE

A fault tolerant reconfigurable system which can be implemented at a reduced hardware and time cost on any SRAM-based FPGA with integrated soft core processors is proposed here. Actual implementation on Xilinx Virtex-5 FPGA relies on an Enhanced Lockstep scheme built using a pair of Micro Blaze cores. To identify the faulty core, a specially designed fault-tolerant Configuration Engine (CE) built using Pico Blaze is proposed. Once the exact error location is determined by a specially designed Scan Motor and a Frame Address (FA) Generator, the error is corrected through partial reconfiguration (PR) combined with roll-forward recovery technique.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

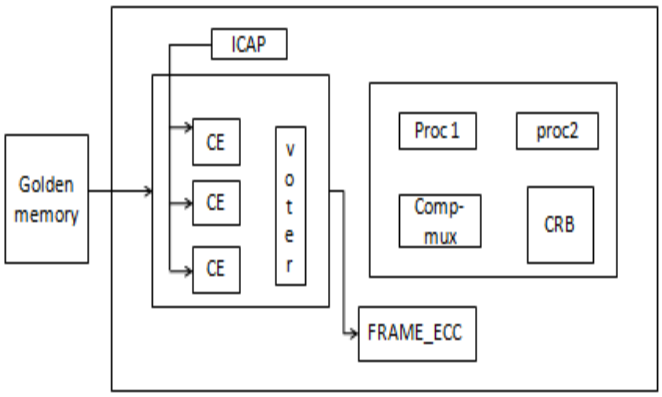


Fig. 1 Fault Tolerant Architecture

VI.STATE RECOVERY PROCEDURE

State Recovery process used here is Roll Forward State Recovery .It recovers and synchronizes the state of two soft core processors. During normal operation(A) ,if a mismatch signal is triggered (B),the Scan process(C)is launched to localize the error. Reconfiguration is carried out in(D),the completion process (E) occurs. The Recovery process is started at (F)and completed at(G).That is Resynchronization process took place in the state(G).

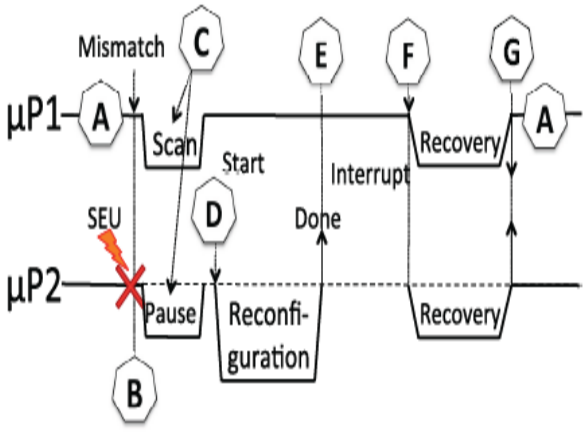


Fig. 2 Roll Forward Error Recovery Scheme

The Recovery process explained detail in the below figure. The interrupt signal is launched by CRB to force each processors to enter into Recovery.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

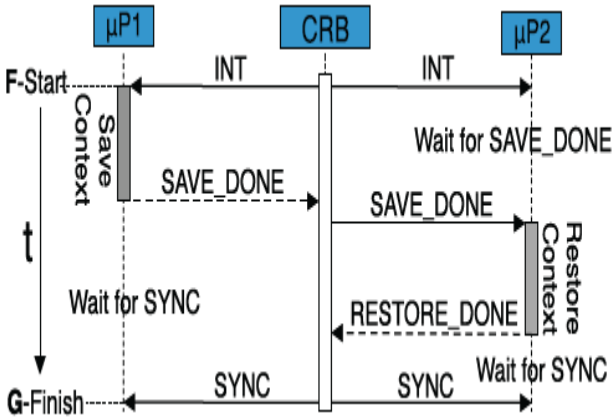


Fig. 3 Recovery process in detail for the enhanced lockstep scheme

VII. VALIDATION THROUGH FAULT INJECTION

Fault injection (also known as fault insertion testing) accelerates the occurrences of faults in a system and the main purpose is to evaluate and debug error handling mechanisms Fault injection is mandatory in safety standard IEC 61508 (adapted by the automotive industry as ISO WD 26262) when claimed diagnosis coverage is at least 90%. As fault injection has become widely used as an experimental dependability validation method, many different techniques for injecting faults have been developed.

Here a fault tolerant configuration engine carries out automatic fault injection campaigns for configuration memory of processor 1 and comp-mux module.Sensitive bits and persistent errors are measured here.Its very difficult to obtain fault injection results for BRAM of CRB.Since the contents of BRAM varies during the operations ,sensitivities and persistence also varies.Here BRAM is protected against errors by ECC.So faults are not injected to BRAM and CRB of processor 1.In fault injection procedure bit flips are injected using Frame based Reconfiguration through ICAP.

Processor1 and comp-mux module occupies 604 and 72 configuration frames respectively.Inorder to evaluate the sensitivity of configuration bits and persistence error of the lockstep scheme single bits are flipped randomley.Upon injection of faults the lockstepped pair of processors executes the application program and checks whether peripherals of softcore processors works correctly.The procedure flow depends on whether comp-mux detects any mismatch during this period.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

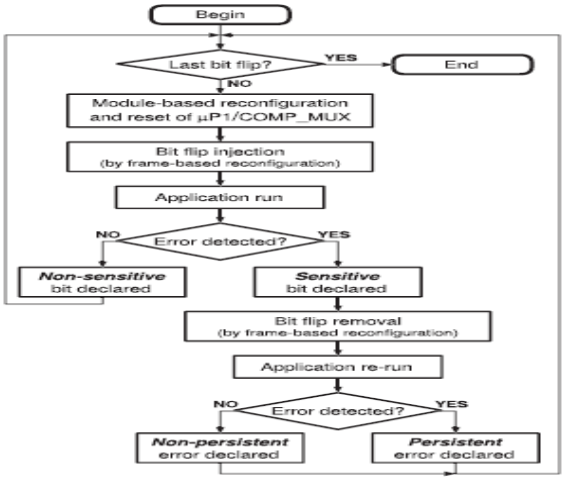


Fig.4 Flow diagram of Fault Injection Process

Step1-If there is no mismatch found the bit is nonsensitive and fault injection for that bit is terminated.Module based reconfiguration is performed to eliminate any undetected persistent errors.

Step2-Next bit is taken and if mismatch detected means the bit is declared sensitive and hence application run is stopped and determines whether this bit causes persistent error or not.Frame based reconfiguration is being performed.then the same application is run and if it again triggers any mismatch signal then module based reconfiguration is performed to correct error and make the FPGA ready for the next bit.

Module	Slices	Utilization	Injected bit flips	Sensitive bits	Sensitivity	Persistence
Processor 1	560	93	39360	3165	8.6	2.3
Comp-Mux	80	83	3936	83	2.1	0.4

Table1: Results of Fault Injection

The table shows Results for fault injection implemented on Virtex 5 FPGA. The experiment reveals that only 8.6 and 2.1 percentage of configuration bits of processor 1 and Comp-Mux are sensitive. Few sensitive bits actually causes persistent errors 2.3 and 0.4 for Processor 1 and Comp-Mux respectively. Based on the results the average interrupted duration caused by the error is around 23 microseconds. The basic enhanced scheme if used will take 6ms to complete .Hence the lockstep scheme proposed here causes less time overhead compared to basic lockstep scheme.

VIII. RESULTS AND DISCUSSION

The system is implemented using Xilinx Virtex-5 FPGA and Xilinx design suite V13.2.Various screen shots obtained is shown below. Fault detection and elimination is being performed. Here once an error is identified it is removed immediately.The dual Lockstep Microblaze cores are operated at 125MHz ie one cycle takes 8ns.
The Picoblaze Cores inside the specially designed engine is runs at 60 MHz ie one cycle takes 16.7ns.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

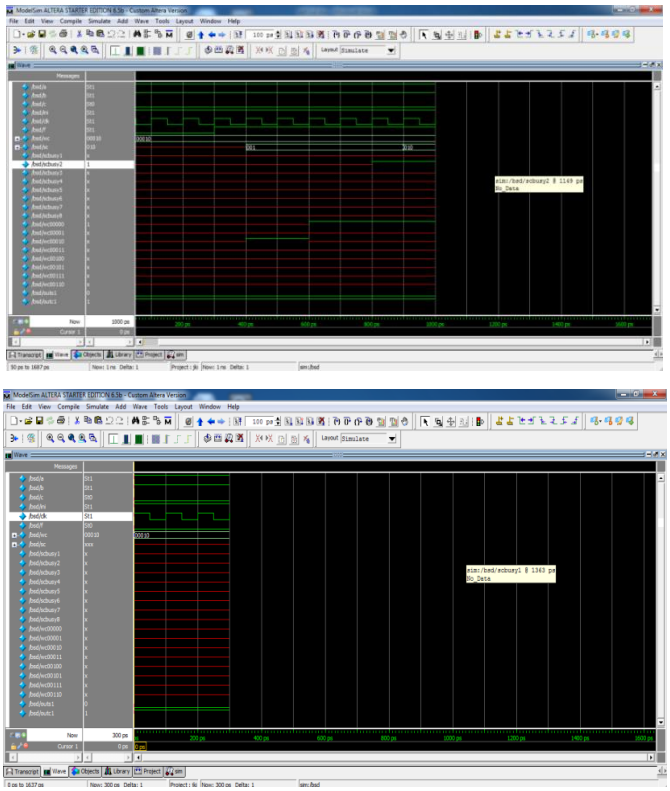
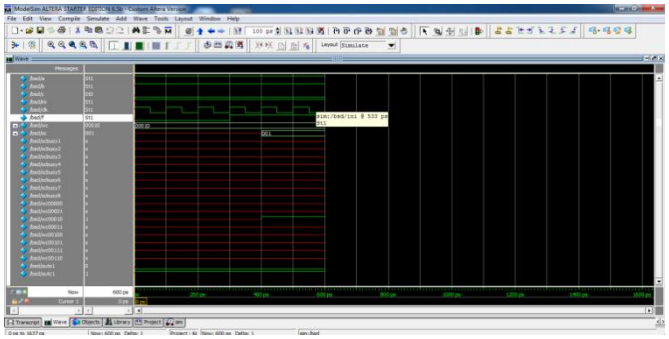


Fig. 5 Partial Reconfiguration

Fig.5 represents how the input is applied for the process of partial reconfiguration . Here full process is not reconfigured.Only part of the FPGA is underwent through the process of Reconfiguration.



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

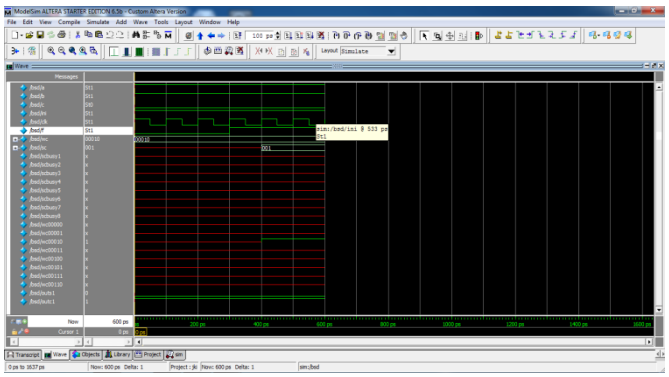


Fig. 6 Roll forward Error Recovery Scheme

Fig.6 represents Roll Forward Error Recovery Scheme. Fault detection is being performed here .Here regular context saving is not needed.If an error is occured it is corrected by copying correct state from fault free processor.It offers better performance.

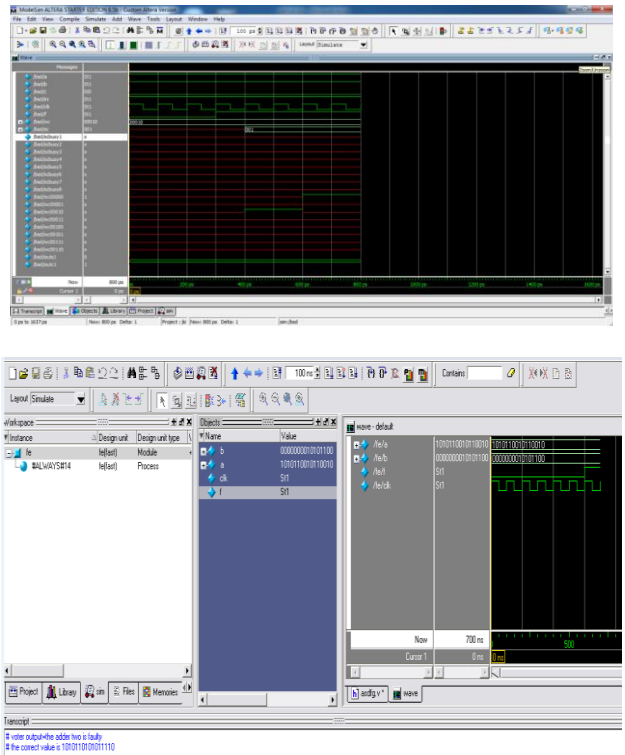


Fig. 7 Fault detection and Correction

Fig.7 shows how fault injection process is being performed.The Faults are injected for the verification process.If a module is reported to be faulty it is corrected and fault free output is displayed .

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

IX.CONCLUSION

This paper proposed a new architecture of a fault-tolerant reconfigurable system which can be implemented on any SRAM-based FPGA with integrated softcore processors. The Repair process is performed using Partial Reconfiguration . Roll forward Error Recovery is used here instead of Roll Backward error recovery schemes in earlier implementations. As a result, the problem of fault latency is alleviated, because faults are detected immediately, once they cause an error. To avoid failures in configuration engine it built using picoblaze core and implemented as fault-tolerant using triple modular redundancy (TMR). The scheme proposed is a valuable alternative to other fault-tolerant softcore schemes. Fault injection experiments are used for validation process. Since Virtex-6 devices also contains FRAME_ECC and ICAP primitives the proposed system can be applied to them also.

REFERENCES

[1] F. Abate et al., "New Techniques for Improving the Performance of the Lockstep Architecture for SEEs Mitigation in FPGA Embedded Processors," IEEE Trans. Nuclear Science, vol. 56, no. 4, pp. 1992-2000, Aug. 2009.
[2] Y. Ichinomiya et al., "Improving the Robustness of a Softcore Processor against SEUs by Using TMR and Partial Reconfiguration," Proc. IEEE Ann. Int'l Symp. Field-Programmable Custom Computing Machines, pp. 47-54, May 2010.
[3] M. Lanuzza et al., "An Efficient and Low-Cost Design Methodology to Improve SRAM-Based FPGA Robustness in Space and Avionics Applications," Proc. Int'l Workshop Reconfigurable Computing: Architectures, Tools and Applications, vol. 5453, pp. 74-84, 2009.
[4] S.-F. Liu et al., "Increasing Reliability of FPGA-Based Adaptive Equalizers in the Presence of Single Event Upsets," IEEE Trans. Nuclear Science, vol. 58, no. 3, pp. 1072-1077, June 2011.
[5] B. Pratt et al., "Fine-Grain SEU Mitigation for FPGAs using Partial TMR," IEEE Trans. Nuclear Science, vol. 55, no. 4, pp. 2274-2280, Aug. 2008.
[6] Xilinx, Inc., "Virtex-5 FPGA Configuration User Guide (UG191 v3.6 ug191.pdf, 2009.
[7] Xilinx, Inc., "Two Flows for Partial Reconfiguration: Module Based or Small Bit Manipulations (XAPP290)," 2002.
[8] K. Morgan et al., "SEU-Induced Persistent Error Propagation in FPGAs," IEEE Trans. Nuclear Science, vol. 52, no. 6, pp. 2438-2445, Dec. 2005.
[9] Xilinx, Inc., "Correcting Single-Event Upsets through Virtex Partial Configuration,"Appl. Note XAPP216 v1.0, support/documentation/application_notes/xapp216.pdf. June 2000.
[10] Xilinx, Inc., "Virtex-5 FPGA User-Guide,"UG190, v4.5, p. 383, Jan. 2009.
[11] E. Fuller et al., "Radiation Testing Update, SEU Mitigation, and Availability Analysis of the Virtex FPGA for Space Re-Configurable Computing," Proc. Int'l Conf. Military and Aerospace Programmable Logic Devices (MAPLD), Sept. 2000.
[12] D.K. Pradhan and N.H. Vaidya, "Roll-Forward and Rollback Recovery: Performance-Reliability Trade-Off," IEEE Trans. Computers, vol. 46, no. 3, pp. 372-378, Mar. 1997.
[13] H.-M. Pham, S. Pillement, and D. Demigny, "A Fault-Tolerant Layer for Dynamically Reconfigurable Multi-Processor Systemon- Chip," Proc. Int'l Conf. Reconfigurable Computing and FPGAs, pp. 284-289, Dec. 2009.
[14] Xilinx, Inc., "MicroBlaze Processor Reference Guide (UG081 v10.3),"http://www.xilinx.com/support/documentation/sw_ manuals/mb_ref_guide.pdf, 2009.