Notice of Violation of IEEE Publication Principles

"A Novel Co-Design Approach for Soft Error Mitigation for Embedded System"
by C. Amutha, M. Ramya, C. Subashini
in the Proceedings of the International Conference on Emerging Trends in Electrical Engineering and Energy Management (ICETEEEM), December 2012, pp. 267-270

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains large portions of text and figures copied verbatim from the paper cited below. The original text was copied without attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following article:

"Compiler-Directed Soft Error Mitigation for Embedded Systems"
by Antonio Martinez-Alvarez, Sergio A. Cuenca-Asensi, Felipe Restrepo-Calle, Francisco R. Pinto, Hipolito Guzman-Miranda, Miguel A. Aguirre
in the IEEE Transactions on Dependable and Secure Computing, Vol 9, No. 2, March 2012, pp. 159-172

A Novel Co-Design Approach for Soft Error Mitigation for Embedded System

C. Amutha
Student, S.A. Engineering College
amuharipri@gmail.com

M. Ramya
Senior Lecturer, S.A. Engineering College
ramya6045@yahoo.co.in

C. Subashini
Assistant Professor, S.A. Engineering College
subashini200235617@yahoo.co.in

***Abstract*—The protection of processor-based systems to mitigate the harmful effect of transient faults. This paper proposes an Depth packet inspection methodology for facilitating the design of fault tolerant embedded systems, the packet inspection is possible in compressed data and thereby achieve high fault coverage in accuracy and speed.The methodology is supported by an infrastructure that hardware and software soft errors mitigation techniques in order to best satisfy both usual design constraints permits to easily combine hardware and software dependability requirements. It is based on a FPGA architecture that facilitates the implementation of software-based techniques, providing a uniform isolated from target hardening core that allows the automatic generation of protected source code.**

***Keywords*: Fault tolerance, soft error, single event upset –SEU, hardware software co design.**

I. INRTODUCTION

Fault tolerance is one of major requirements for Embedded systems. As the embedded systems become more complex, there is more chances for various failures. When designing Embedded system has to deal with the faults. Before dealing with faults it has to identify and understand the types and nature of faults. Faults can be hardware and software.The fault occur inside a processor will be considered as hardware fault and error occurs externally due to instruction set is considered as soft error. Hardware faults cause errors that don't go away for instance reseting of MCU doesn't restore from fault condition. Software errors can occur due to transient. Ever increasing miniaturization of electronic components has led to important advances FPGA susceptible to transient faults induced by radiation performance. [1], [2]. These intermittent faults do not provoke a permanent damage, but may result in the incorrect execution of the program by altering signal transfers or stored values, these faults are also called soft errors [3]. Although these faults are more frequent in the space environment, they are also present to in the atmosphere [4] and even at ground level [5], [6]. In the proposed work fault tolerance achieved by depth packet inspection. Depth Packet Inspection called complete packet inspection Information extraction is a form of computer network packet filter in that examines the data part and possibly also the header of a packet as it passes an inspection point, searching for protocol non-compliance, intrusions, or defined criteria to decide whether the packet may pass or needs to be routed to a different destination and statistical information. Depth packet Inspector can handle compressed data. The proposed work is simulated by using model sim software and VHDL language used.

Fig 1.The Block diagram description in fault tolerant embedded design are.

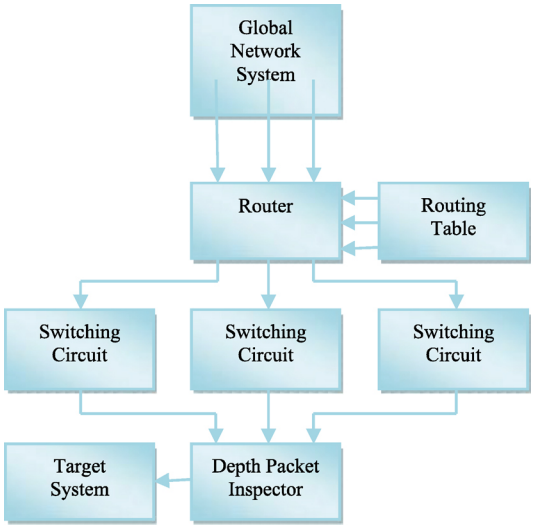

Fig. 1: Block Diagram of Fault Tolerant Embedded System Basic Design

II. FAULT TOLERANT EMBEDDED SYSTEM BASIC DESIGN

The proposed system consist of six modules are depth packet inspector, interface system, fault tolerant monitor, memory, router, Embedded core. This project uses FPGA Cyclone IV as processor. Global network is a multiple system. Data from multiple system routed to switching circuit. Switching circuit switch the data to Depth packet inspector. In depth packet inspector fault tolerant provided for fault data. After providing fault tolerant, the data flow to target. The target is an Embedded Core to which all peripheral, Interface

System, Memory connected. Fault tolerant monitor and controller monitor all data flow in the peripheral, memory, embedded core, interface system. Fault data provided tolerant by Fault tolerant monitor and controller before passing to target. A router is specialized computer connected to more than one network running software that allows the router to move data from global nemtwork system to switching circuit. Data given input o packet analyzer . Configuration given by packet analyzer to forward data to destination. Data packet given to construct character table. Split the character from packet given to scan. Scan will identify the fault and prime solver provide fault tolerant .Depth packet inspection is a form of computer network packet filtering that examine the data part of the packet as it passes an inspection point. Protocol converter matches input CPU Protocol and output Peripheral Protocol. Arbiter decides to which peripheral to connect. Embedded Core consist processor FPGA CycloneIV

III. Fault Terminology

A Single Event Upset Fault

A single event upset (**SEU**) is a change of state caused by ions or electro-magnetic radiation striking a sensitive node in a micro-electronic device, such as in a microprocessor, semiconductor, memory or power transistors. Terrestrial SEU arise due to cosmic particles colliding with atoms in the atmosphere, creating cascades or showers of neutrons and protons, which in turn may interact with electronics.

B Silent Data Corruption Fault

The silent data corruption faults are the most dangerous errors as there is no indication that the data is incorrect. The error affects the expected output.

C Hang Fault

Fault causes the program to abnormally finish its execution or to remain forever in a infinite loop.

D Stuck at Fault

Individual signals and pins are assumed to be stuck at Logical '1', '0 '. An output is tied to a logical 1 state during test generation. Likewise the output could be tied to a logical 0 to model the behavior of a defective circuit that cannot switch its output pin.

E Sequential Fault

In sequential fault diagnosis the process of fault location is carried out step by step, where each step depends on the result of the diagnostic experiment at the previous step. Such a test experiment is called adaptive testing.

IV. DepthPacket Inspection Algorithm

1. Generate random data packet.
2. Apply a random distribution to choose between (0-4) seed.
3. Random result zero ,no compression.
4. If compression present 1-zip,2-rar,3-tar,4-iso.
5. Add Header according to the compression applied.
6. Receive input packet
7. Scan packet header.
8. Jump to Testing if no compressiontook place.
   i. Check option 1 zip compression.
   ii. Check option 2 rar compression.
   iii. Check option 3 tar compression.
   iv. Check option 4 iso compression.
1. After decompression pass to testingblock.
2. Testing:
   Single Event Upset Fault
   Hang Fault
   Silent Data Corruption
   Stuck at Fault
   Sequence Fault
   END

A Bloom Filter

Fig2. Bloom filter scans the streaming data and checks the strings of corresponding length. Whenever a Bloom filter detects suspicious string, an analyzer probes this string to decide whether it indeed belongs to the given set of strings or is a false positive.

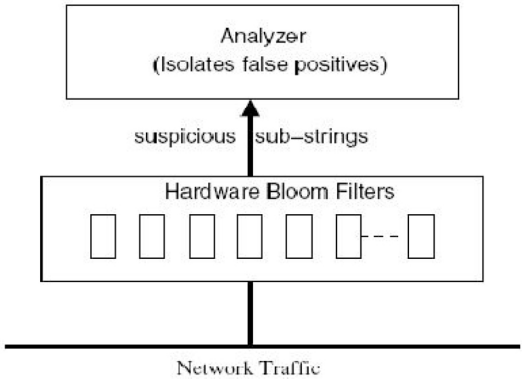

Fig. 2: Bloom Filter

The false-positive probability expression for the Bloom filter,

f = (1/2) [(M/G)/N] ln2, Throughout: R4= 3.2 / (1,920f + 1.019)

After substituting the value of f into the expression for R4 and plotting throughput RG for a total of N = 10,000 strings obtain the graph in Fig 3. The graph shows results for two values of *p*, the probability of a string's true occurrence of strings tuned the system for a total of *N* = 10,000 strings, each of B = 24 distinct

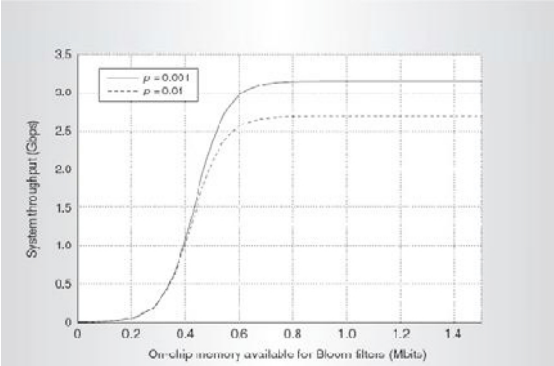

Fig. 3: System Throughput

V. Software Hardening Environment

The main features of SHE can be expressed as:

.Flexible: that is, easy to extend its hardening Capabilities. Re-targetable output: to provide reusing of code.

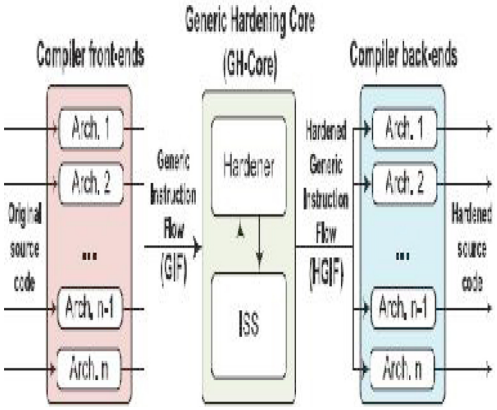

Fig. 4: Software Hardening Environment

The proposed environment establishes a complete tool for the fault-tolerant software development, allowing the design and implementation of software-based mitigation Techniques, which can be automatically applied into programs. The environment is made up of a target compiler Hardener, an Instruction Set Simulator (ISS), and depth packet inspector.

TABLE 1: Registers Usage for FIR

Register	# Writes	#Reads	#Read/ Write	Lifetime [%]
0	2806	3320	0	34.81%
1	1785	765	8160	75.16%
2	1276	1531	10841	88.27%
3	1276	1530	8925	88.27%
F	1	256	255	100.0%

Table1 presents the information about registers usage (number of accesses and lifetime) for the FIR program (this program only uses the following registers: 0, 1, 2, 3, and F). Registers lifetime is expressed as the percentage of the total program time.

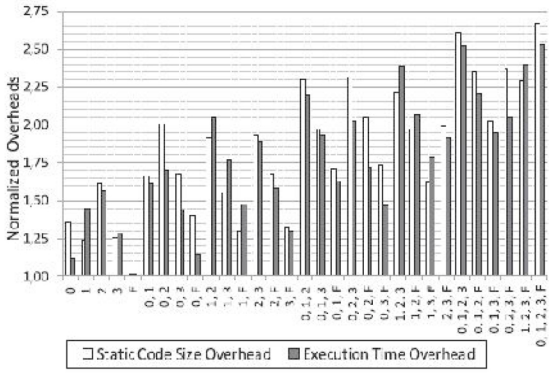

Fig. 5: Normalized Static Code size and Execution Time Overheads

Fig. 5 Normalized static code size and execution time overheads for FIR selective hardened versions. Overhead results for several selectively Hardened register subsets using software protection. Vulnerability of each register depends on the lifetime of the Register during the program execution.

B Fault Classification

Fig.6 shows the fault classification percentages and the normalized MWTF obtained for each in the FTUnshades. Results have been classified in the with the ISS evaluation.

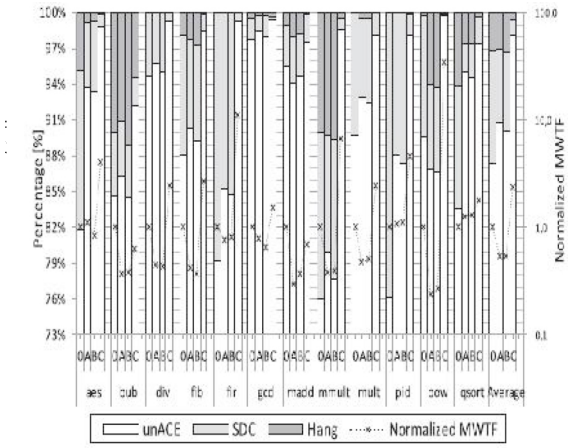

Fig. 6: Fault Classification Percentages

VI. Experimental Results

The tool developed for hardening five circuits, which mapped on a Xilinx Cyclone IV device. To evaluate the robustness of the circuit obtained through Depth Packet Inspection Algorithm against transient faults affecting the FPGA's configuration memory. The fault injection environments are presented in Table 2. Where Injected Faults is the number of SEUs and Wrong Answer is the number of SEUs for which the

faulty circuit produces outputs that differ from the fault-free one. Depth Packet Inspection (DPI) provides high fault tolerant capability. The proposed work is simulated using Model sim software using VHDL language.

TABLE 2: FAULT INJECTION RESULTS

Circuit	Injected Faults [#]		Wrong Answers [#]					
			Plain version		TMR version		RoRA version	
	CLB	Routing	CLB	Routing	CLB	Routing	CLB	Routing
Add8	2,558	12,442	2,550	12,037	97	1,255	29	1
Add16	2,410	12,590	2,408	12,190	83	1,609	37	4
Mul8	2,440	12,560	2,390	12,213	91	1,886	20	3
Filter	2,427	12,573	2,398	12,244	86	1,895	39	5
CAN	2,550	12,450	2,545	12,404	71	2,005	38	8

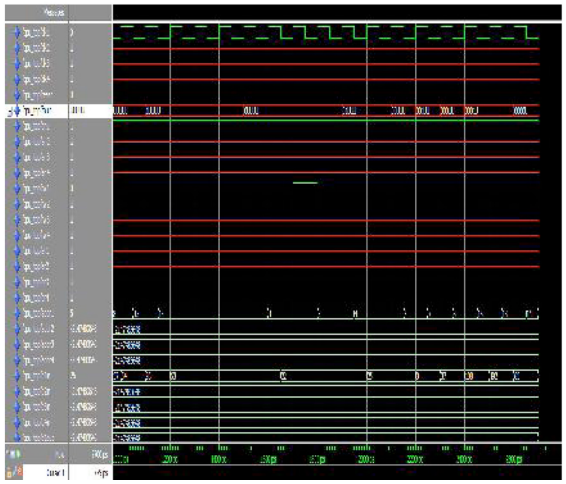

Fig. 7: Fault Tolerant Processor Simulation

Fig 7 shows fault tolerant processor consist of 4 pins. Fourth pin indicate single event upset fault which end the execution of processor, after fault tolerant the execution of processor will be restarted. Third pin indicate hang fault which stops the processor without changes. Second pin indicate stuck at 0 fault, after fault tolerant data available in the output. First pin indicate the silent data corruption fault, after fault tolerant Correct data available in the output.

VII. CONCLUSION

The embedded system must be reliable if a fault occurs it provide faulty output thereby affect accuracy of the results, damage equipment, or waste expensive resources. This paper presents Depth Packet Inspection methodology that is able to guide the co design of fault-tolerant hardware and software systems. It is supported by an infrastructure that facilitates the exploration of the design space between hardware and software soft-errors mitigation strategies. As a result, this new methodology suggests the implementation of automatic hardening tasks within the presented platform and opens up interesting new boundaries in the design space of embedded system.

REFERENCES

[1] R. Baumann, "Radiation-Induced Soft Errors in Advanced Semiconductor Technologies," IEEE Trans. Device and Materials Reliability, vol. 5, no. 3, pp. 305-316, Sept. 2005.

[2] P. Shivakumar, M. Kistler, S.W. Keckler, D. Burger, and L. Alvisi, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic" Proc. Int'l Conf. Dependable Systems and Networks, pp. 389-398, 2002.

[3] T. Karnik, P. Hazucha, and J. Patel, "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 2, pp. 128-143, Apr.-Jun. 2004.

[4] R. Edwards, C. Dyer, and E. Normand, "Technical Standard for Atmospheric Radiation Single Even t Effects (SEE) on Avionics Electronics," Proc. IEEE Radiation Effects Data Workshop (REDW '04) pp. 1-5, 2004.

[5] R. Baumann, "Soft Errors in Commercial semiconductor Technology: Overview and Scaling Trends," IEEE 2002 Reliability Physics Symp. Tutorial Notes, Reliability Fundamentals, pp. 121-01.1–121 01.14, IEEE Press, Apr. 2002

[6] S.E. Michalak, K.W. Harris, N.W. Hengartner, B.E. Takala, and S.A. Wender, "Predicting the Number of Fatal Soft Errors in Los Alamos National Laboratory's ASC Q Supercomputer," IEEETrans. Device and Materials Reliability, vol. 5, no. 3, pp. 329-335,Sept. 2005

[7] M. Pignol, "COTS-Base applications in Space Avionics. In EDDA, editor," Proc. 13th Design, Automation and Test in Europe conf., (DATE '10), p. 1213, Mar. 2010.

[8] V.K. Reddy, S. Parthasarathy, and E.Rotenberg,"Understanding prediction-Based Partial Redundant Threading for Low-Overhead, High-Coverage Fault Tolerance," ACM SIGPLAN NOTICES,vol. 41, no. 11, pp. 83-94, Nov. 2006.

[9] R. Naseer , R.Z. Bhatti , and J. Draper, "Analysis of Soft Error Mitigation Techniques for Register Files in IBM Cu-08 90nm Technology," Proc. 49th IEEE Int'l Midwest Symp. Circuits and Systems, pp. 515-519, Aug. 2006.

[10] H. Guzman-Miranda, M.A. Aguirre, and J. Tombs, "Non invasive Fault Classification,Robustness and Recovery Time measurement in Microprocessor Type Architectures Subjected to Radiation- InducedErrors,"IEEE Trans. Instrumentation and Measurement, vol. 58, no. 5, pp. 1514-1524, May. 2009.