Design and Analysis of Transient Fault Tolerance in SRAM with different NT Techniques

S. Ravichand
Research Scholar, JNTUK,
Kakinada, Andhra Pradesh.

T. Madhu, PhD
Professor in ECE,
Swarnandhra Inst. Of Engg.,
and Tech., Narasapur,
Andhrapradesh

M. Sailaja, PhD
Professor in ECE, JNTUK,
Kakinada, Andhra Pradesh,
India.

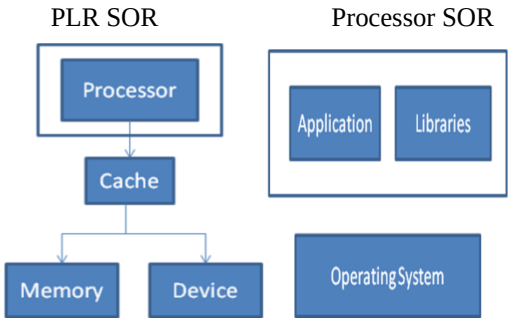
ABSTRACT
In digital domain applications one of the main important analyses is to check the system performance even if the fault is occurred. This paper describes the fault tolerance technique based on the software approach for SRAM memory unit present in the multi core system using Processor Level Redundancy (PLR). The PLR proceeds with software-centric approach, soft fault tolerance which ensuring a correct software execution. In this approach we applied here only for SRAM available in the processor. This scheme is used at user space level which does not necessitate changes to the original application. In this approach is used to detect the soft errors presented in the memory unit and it will recover from the fault without stop the process of the memory unit. To design the SRAM here we use the different nT technique. The main goal of this approach is which implements fault error detection and fault recovery mechanism to check the performance of the memory unit. This paper presents software based nT for SRAM design and analysis transient fault tolerance using Process Level Redundancy (PLR) is lower when comparable to existed.

Keywords
SRAM, Fault Tolerance, Process Level Redundancy, nT design

1. INTRODUCTION
Transient faults, also called as soft errors, which concern in the reliability of computer systems [1]. A transient fault occurs even in the presence of error. Fault cause error results failure. Fault cause error observed by deviation from expected behavior results failure. It occurs event (e.g., cosmic, device coupling) alleviation or removal of sufficient charge to invert or change the state of the transistor. The inverted value may show effect on program under execution. Current trend in process technology shows that the error rate will remain comparatively constant in future [2]. The number of usable transistor per chip continuously grows in an exponential manner, which increases dramatically. These trends had shown that to ensure correct execution operation of systems. Transient fault characteristics are availability, accessibility, dependability and reliability. The memory is easily protected with error correcting code and parity within high performance microprocessor [3]. However, the same specified techniques cannot be adopted for general purpose computing domain directly.

Application specific constraints Fault results in a glitch which user may not be noticed. Then, the reliability improves to meet user requirements of failure rates. While software technique cannot render a reliability level of hardware technique, they significantly provides a low cost and flexible (zero hardware design cost).existing software transient fault tolerance approaches use the encyclopedias to insert redundant instructions for checking computation and control flow process. Redundant more than is needed, desired or required to ensure correct execution [4].

2. LITERATURE SURVEY
Recent work has demonstrated the use of transient fault which describes the service even in the presence of fault. It describes a low overhead software based approach [5-6]. Fault can be classified by benign fault, Silent Data Corruption (SDC) and Detected Uncoverable Error (DUE) [7]. Benign fault is a transient fault that does not propagate to affect the correctness of an application. Many benign faults that propagate without affecting program correctness can be safely ignored [8]. A transient fault that is undetected and propagates to corrupt program output is considered as Silent Data Corruption SDC. Detected Uncoverable Error (DUE).A transient fault which is detected and propagates without possibility of recovery is considered a DUE.



(a) Hardware-Centric **(b) Software Centric**

Figure.1. Hardware and Software-Centric Transient Fault Detection Models

Most previous access is hardware-centric-even compiler approaches. Swift software implemented fault tolerance technique is used. Software Centric is able to leverage the strength of a software approach. PLR improves performance over existing software transient fault tolerance techniques and moves ahead to make it possible for software fault tolerant solutions with comparable performance to hard-ware techniques [9-10]. In Fig.1, Transient fault provide a service even in the presence of error. Fault causes error results in failure.SOR concept are used for defining the boundary of reliability in redundant hardware design [11-13].

3. METHODOLOGY
Many problems become increasingly large and complex, forcing the use of digital system design for their efficient and effective applications [14-15]. Here the circuit design consists of SRAM block, shift register, adder, and master and slave process. Top module consist SRAM which is used to store the data. Shift register we used totally are three shift register, they are group of flip flops cascaded in chain. All the flip flops are driven by a common clock. Next stage is master and slave

process we are designing adder circuit. Watch dog timer which is used to detect the error here designed as a NOR and inverter. Shift register we used totally three shift register they are group of flip flops are connected in a chain. All the flip flops are driven by a common clock. Next stage is master and slave process we are designing adder circuit. Watch dog timer which is used to detect the error here designed as a NOR and inverter. The schematic design Fig.2 is shown below with three redundant processes. Thus, the reliability improves to meet user expectations of failure rates. This is mainly used to leverage overhead mechanism.

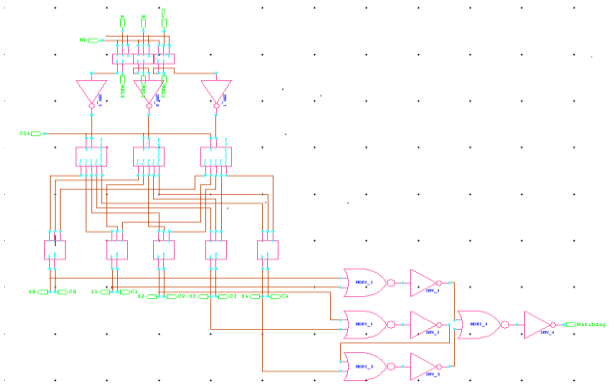


Figure.2. PLR Circuit of Transient Fault Design

Now we are designing memory in that memory design input signal are given to memory to store the memory value with respect to word line WL. If WL high mean the data will be positive in shift register module or else it will be store as it is. Shift register consists of D flip flop with five bit. Based on input memory will produce the output with respect to clock signal. From the output of shift register is given to master and slave module it will function the full adder. Full adder process designed by using a two half adder.

SRAM reveal data reminisce, but it is still volatile in the conventional sense that data is eventually lost when the memory is not powered. Static memory which is mainly used for a storage device it is a permanent storage. Shift register are sequential logic circuit used for storing binary digits. They are group of flip-flops connected in a chain. So that the output received from one flip-flop becomes the input of next flip-flop. A common clock drives all the flip-flops and all are set or reset simultaneously. One of the redundant processes assigned to be the master process and the other one as the slave processes. Here we used a adder application here two half adder as designed as a full adder which propagates the sum and carry output. Modeling of the memories is needed to optimize power, delay and area in SRAM design and to characterize the behavior of the SRAM and help making design decisions before running SPICE simulations. There have been many proposed models and tools developed to predict the SRAM performance. However, these models and tools are all based on traditional 6T, 6.5T, 8T and 10T SRAM design.

4. RESULTS AND DISCUSSIONS

4.1 Shift Register

In Fig.3, shift register are a type of sequential logic circuit mainly used for storage. They are group of flip-flops connected in a chain. So that, the output received from one flip-flop becomes the input of the next flip-flop. All the flip-flops are driven by common clock and all are set or reset simultaneously. SISO are connected in achain one of the flipflop output becomes the input to the next flip flop which are driven in acommon clock.it is mainly used for the storage device consists of number of register.

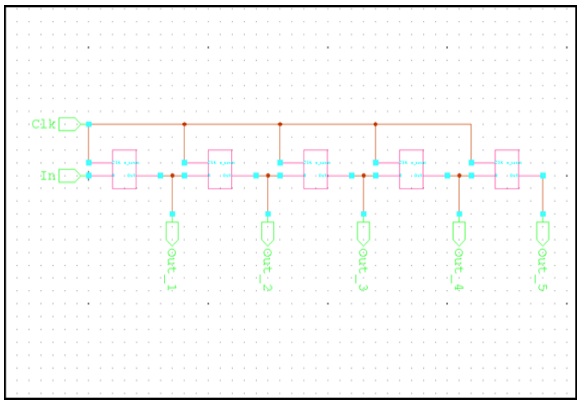


Figure.3.Shift Register design

4.2 Design Of Full Adder Using Two Half Adder

In Fig.4, the structure of full adder using two half adder it consist of sum and carry output.It consists of 3 input and two output.

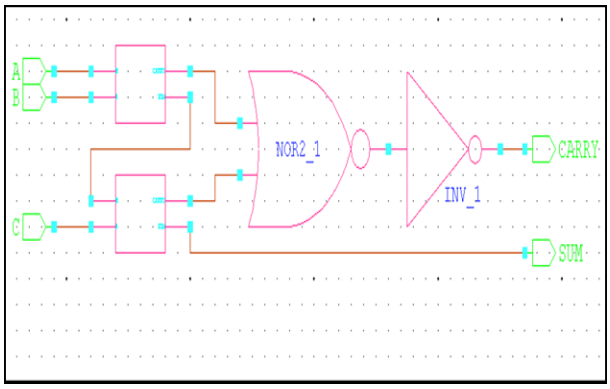


Figure.4. Structure Of Full Adder Using Two Half Adder

4.3 6T SRAM BASE FULL DESIGN

Here to design the PLR circuit for transient fault design. This circuit consists of four stages1) Memory block 2) shift register 3) master and slave process 4) watch dog timer. In Fig.5, is schematic of 6T SRAM. The simulated output of fault free and faulty outputs are shown in Fig 6 and 7.

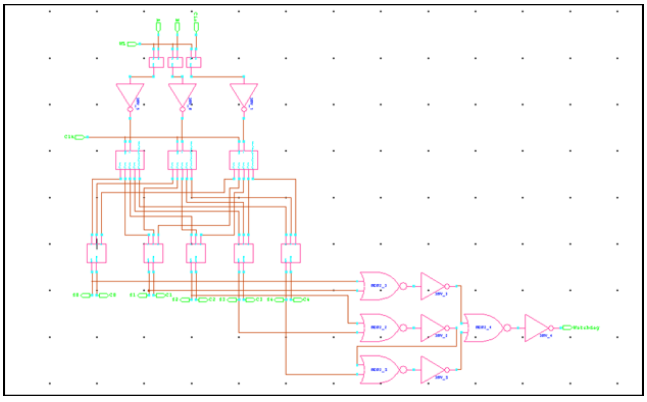


Figure 5.Structure of 6T SRAM Based Full Schematic

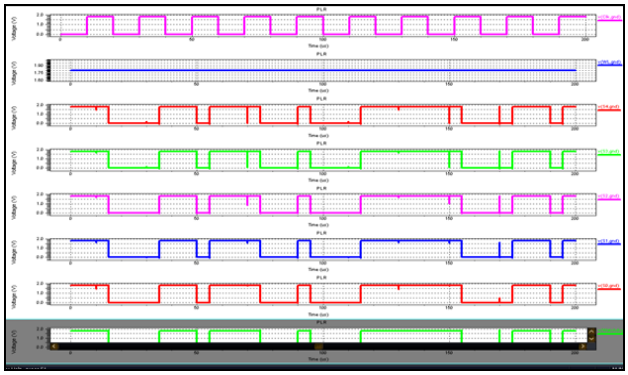


Figure 6.Simulated Output For Fault Free Output

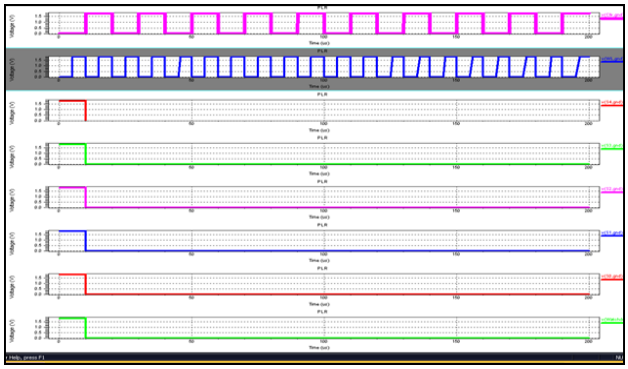


Figure 7.Simulated Output For Faulty Output

4.4 6.5T SRAM CELL

The basic architecture of the proposed write read decoupled SRAM block illustrated in Fig.8. The number of transistors in each block depends on the number of bits stored in the block. The average-6.5T block holds sixteen bits through sixteen back-to-back connected inverters the storage nodes are connected to the local bit-lines (LBL and LBLB). These local bit-lines are decoupled from the write bit-lines (WBL and WBLB*)* via Mwr1–2 during a write operation and from the global read bit-lines (RBL and RBLB*)* via Mrd1–4 during a read operation. This new write/read-decoupled (WRD) technique allows complete isolation of these sixteen bits. The schematic consists of 6.5T SRAM and the average-6.5T block holds sixteen bits through sixteen back-to-back connected inverters, shift register, and adder master and slave process. The simulated output of fault free and faulty outputs are shown in Fig. 9 and 10.

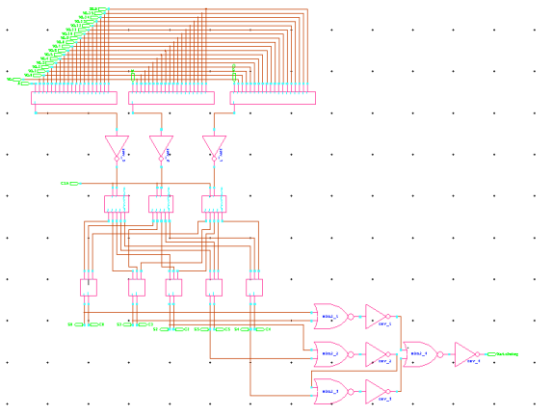


Figure 8. Base Full Schematic For 6.5T SRAM

Figure 9. Fault Free output for 6.5T SRAM

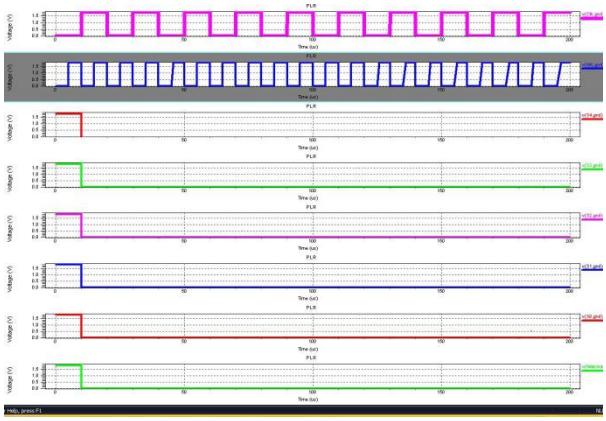


Figure 10. Faulty output of 6.5T SRAM

4.5 Base Full 8t Sram Schematic

The design of an 8T SRAM cell (Shown in Fig.11), at the start of the read cycle the read bit line is pre charged to full swing voltage. After the pre charge phase is over, Read Word line (RWL) is asserted that drives the access transistor M5 ON. The schematic is based on the structure of 8T SRAM based base full schematic which consist of memory block,shift register,redundand process of master and slave process. The simulated output of fault free and faulty outputs are shown in Fig. 12 and 13.

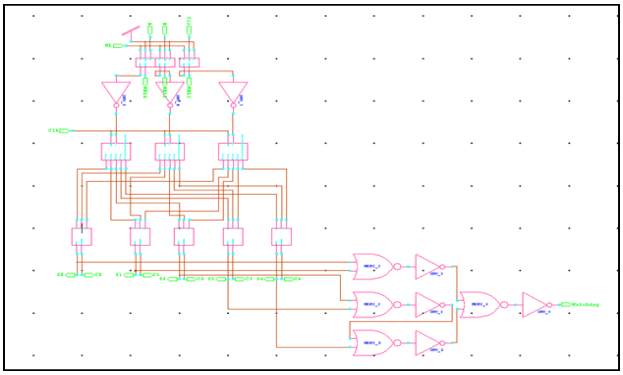


Figure 11. Structure of 8T SRAM Based Full Schematic

Figure 12.Simulated output for Fault free output

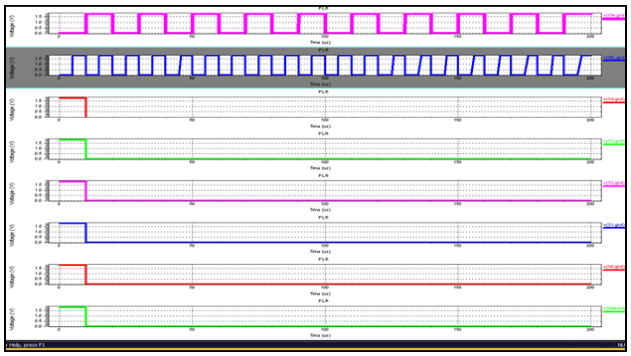


Figure 13.Simulated of Faulty Output

4.6 10t Sram Base Full Schematic

In Fig.14, 10T SRAM is similar to the 8T SRAM with extra transistors are connected to the read decoupled path with bit lines (BL & BLB) to avoid the draining of read current to ground. The architecture of the 10T cell is similar to the single ended 8T cell. In addition to the 8 transistors, 2 transistors are added in the decoupled read path. The simulated output of fault free and faulty outputs are shown in Fig. 15 and 16.

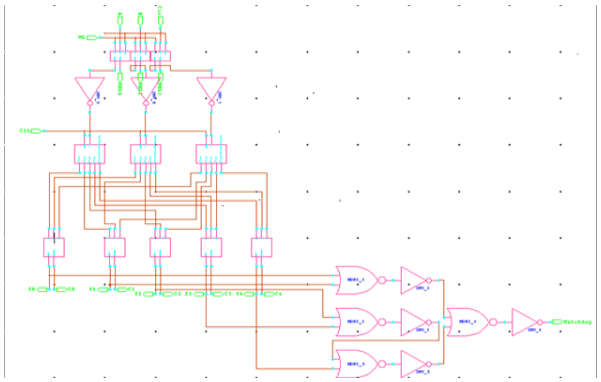


Figure 14. Structure of 10T SRAM Based Full Schematic

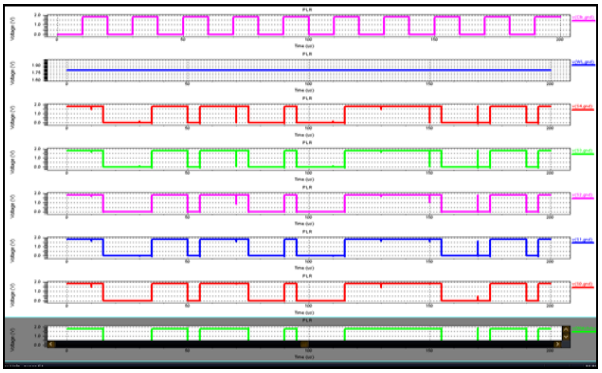


Figure 16. Simulated output of Fault free output

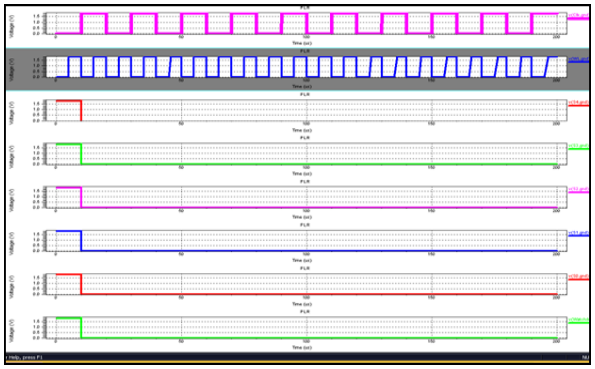


Figure 17.Simulated output of Faulty output

4.7 POWER ANALYSIS AND OPTIMIZATION

From this power result we also calculate the Power Delay Product ((PDP=average power consumed * propagation time delay), in fJ (10^-15))

Static current I=P/V (mA)

Operating frequency=1/T HZ $=1/T_{On}+T_{Off}$

To calculate the Area of Transistor:

Area of transistor=No. of transistor*length * width (μm^2)

Throughput=operating frequency*No. of bits at output (bits/sec)

SRAM is semiconductor memory cell. It stores one bit of information. It is faster and consumes very less power as compared to other memory cells. SRAM is vital component in a chip or microprocessor IC. 10T SRAM cell performs better than 6T. The following comparision Table 1 Table 2 shows that the detail about the design of 6T,6.5T,8T and 10T. The optimizatiom is based on the simulation results of 6T,6.5T, 8T ,10T of average power consumed power and delay we calculate the following table from this we optimize the how power consumed and concluded that the 10 T SRAM has advantageous compared to 6T and 8T. The total base full schematic consumed the average power,sttic power,static current,energy delay product be optimized in Table .1 shows that the four parameters comparitive study for 6T,6.5T,8T and 10T.

Table 1: Comparitive Study of Power & Delay For Base Full Schematic

Design	Power	Delay	Power delay product
6T	$1.134*10^{-002}$	$1.700*10^{-004}$	$1.9278*10^{-6}$ ws
6.5T	$3.864*10^{-003}$	$7.505*10^{-005}$	$2.8999*10^{-7}$ws
8T	$1.607*10^{-002}$	$1.0001*10^{-005}$	$1.6071*10^{-7}$ws
10T	$8.637*10^{-003}$	$7.000*10^{-006}$	$60.459*10^{-8}$ws

μw- Micro Watt , mw- milli watt , mA- milli Ampherem , nws- newton watt second

Table 2: Power Delay Product

Parameters	6T	6.5T	8T	10T
Power average	0.177mw	2.126mw	0.222mw	0.109mw
Static power	11mw	63.4mw	16mw	8.63mw
Static current	6.1mA	35.22mA	8.8Ma	4.7mA
Energy Delay Product	1.9278mws	289.6nws	1.607nws	604.5nws
Operating frequency	25Hz	25Hz	25Hz	25Hz
Area of transistor	$3.75\mu m^2$	$40\mu m^2$	$5\mu m^2$	$6.25\mu m^2$
Throughput	25 bits/sec	25 bits/sec	25 bits/sec	25 bits/sec

5. CONCLUSION & FUTURE WORK

To Remove all glitch in the circuit and to design a fault free circuit. It can be concluded that focuses on a method of charge sharing between the bit lines of 10T SRAM and these method consumes less power for read operations and cut systems dynamic power budget to greater extent. The significant improvements obtained in the results through the use of 10T cell which will be applicable for future low power memory design. Fault injection demonstrates a known fact that PLR's software-centric fault detection model detects faults to an acceptable degree by safely ignoring benign faults. Present a software implemented transient fault tolerance technique to utilize general purpose hardware in SRAM with multi cores. PLR performance meliorates upon existing software transient fault tolerance techniques and takes a step toward enabling software fault tolerant solutions with comparable performance to hard-ware techniques. Applications such as sensor network domain that is battery driven application. In future work we will design a processing unit circuit and study about how to design circuits with multiple outputs.

6. REFERENCES

[1]. Alex Shye, Student Member, IEEE, Joseph Blomstedt, 'PLR: A Software Approach to Transient Fault Tolerance for Multi core Architectures' *IEEE Transactions on Dependable and Secure Computing* (**TDSC**). April-June 2009 (vol. 6 no. 2) pp. 135-148.

[2].Alex Shye Tipp Moseley† Vijay Janapa Reddi Joseph Blomstedt Daniel A. Connors(2013),'Using Process-Level Redundancy to Exploit Multiple Cores for Transient Fault Tolerance' *In proceedings of the 37th IEEE/IFIP International Conference on Dependable Systems and Networks* (**DSN**). Edinburgh, UK. June 25-28, 2007.

[3]. Christopher Weaver1, Joel Emer1, Shubhendu S. Mukherjee1, and Steven K. Reinhardt 'Techniques to Reduce the Soft Error Rate of a High Performance Microprocessor', Proc. Ninth Int'l Conf. Architectural Support for Programming Languages, 1, 2-Jan.2007.

[4]. R. L. Graham, S.-E. Choi, D. J. Daniel, N. N. Desai, R. G. Minnich, C. E. Rasmussen, L. D. Risinger, and M. W. Sukalski."A network-failure-tolerant message-passing system for terascale clusters. In ICS. New York, USA, June. 22-26 2002.

[5]. Hamid Mushtaq, Zaid Al-Ars, Koen Bertels),'Efficient Software-Based Fault Tolerance Approach on Multi core Platforms' Proc. 37th Int'l Conf. Dependable Systems and Networks, DSN '10.

[6]. Iwagaki, T; nakaso, t ohkubo, r; Ichihara, h ,'Scheduling algorithm in data path synthesis for long duration transient fault tolerance', IEEE Reliability Physics Tutorial Notes, Reliability Fundamentals, pp. 121_01.1-121_01.14. Feb. 2009.

[7]. Karthik, Sundaramoorthy, Zach Purser Eric Rotenberg, 'Slipstream processors: improving both performance and fault tolerance', in Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), July .2008.

[8]. S.K. Reinhardt and S.S. Mukherjee, "Transient Fault Detection via Simultaneous Multithreading," Proc. 27th Ann. Int'l Symp. Computer Architecture (ISCA), 2000.

[9]. Karnik, T.; Bloechel, B.; Soumyanath, K.; De, V.; Borkar, S., "Scaling trends of cosmic ray induced soft errors in static latches beyond 0.18 /spl mu/," in *VLSI Circuits, 2001. Digest of Technical Papers. 2001 Symposium on* , vol., no., pp.61-62, 14-16 June 2001doi: 10.1109/VLSIC.2001.934195.

[10]. Kottke, T.; Steininger, A., "A Fail-Silent Reconfigurable Superscalar Processor," in Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on , vol., no., pp.232-239, 17-19 Dec. 2007 doi: 10.1109/PRDC. 2007.16.

[11]. T.N. Vijaykumar, I. Pomeranz, and K. Cheng, "Transient-Fault Recovery Using Simultaneous Multithreading," Proc. 29th Int'l Symp. Computer Architecture (ISCA), 2002.

[12]. N. Oh et al.,'Error Detection by Duplicated Instructions in Super- Scalar Processors', IEEE Trans. Reliability, vol. 51, no. 1.Mar 2002.

[13]. Reis, G. A., Chang, J., Vachharajani, N., Rangan, R., August, D. I.: SWIFT: Software Implemented Fault Tolerance. In: Proceedings of the International Symposium on Code generation and optimization, pp. 243–254. IEEE Press, Washington DC (2005).

[14]. Kulkarni, S.S.; Ebnenasir, A., "Complexity issues in automated synthesis of failsafe fault-tolerance," in *Dependable and Secure Computing, IEEE Transactions on* , vol.2, no.3, pp.201-215,July-Sept.2005 doi: 10.1109/TDSC.2005.29.

[15].Karnik, T.; Bloechel, B.; Soumyanath, K.; De, V.; Borkar, S., "Scaling trends of cosmic ray induced soft errors in static latches beyond 0.18 /spl mu/," in VLSI Circuits, 2001. Digest of Technical Papers. 2001 Symposium on , vol., no., pp.61-62, 14-16 June 2001doi: 10.1109/VLSIC.2001.934195.