MANTECH Publications

Journal of Current Trends in Electrical Engineering
Volume 3, Issue 2, May-August, 2018

Stochastic Checkers Design for Verification of Digital System

Bhargavi B A[1], Dr. Roopa M[2]

Department of Electronics and Communication Engineering[1,2]

Dayananda Sagar College of Engineering Bangalore, India [1,2]

Corresponding Authors' E-mail: *bhargavi24.anand@gmail.com[1], surajroopa@gmail.com[2]*

Abstract

There has been an increasing concern about the growing vulnerability of future computing systems resulting errors in the underlying hardware. Providing reliability unlike conventional fault tolerant techniques, without additional resources is a critical challenge in deeply scaled CMOS and post CMOS era. Hence designing a reliable system with low overheads is very important. To tackle this challenge, we take the benefit of different intrinsic resilience of application domains such as multimedia, recognition, mining, search and analytics which produces acceptable outputs despite occasional approximate computations. In this paper as a new approach of performing approximate error checking at greatly reduced overhead is proposed. Stochastic Checkers are designed by using stochastic computing (SC). This checker has main benefit of using compact arithmetic elements. Additionally it is innately fault tolerant because of its distinctive encoding of numerical values. Hence by using necessary Binary To Stochastic (BTS) converter and Stochastic To Binary (STB) converter error checkers are designed. Further Triple Modular Redundancy[TMR] is proposed to depict the advantage of Stochastic Checker (StoCK), further both the proposed designs are explored and Simulation are carried out for both FIR filter and a equation, applications using Xilinx Vivado design suite, as compared with traditional fault tolerant technique while maintaining high fault coverage.

Keywords— *Approximate error detection, Binary to Stochastic (BTS), Fault Detection, Stochastic Computing (SC), Stochastic to Binary (STB), Triple Modular Redundancy [TMR].*

INTRODUCTION

Scaling toward the boundaries of CMOS, and probably into the post-CMOS era, is anticipated to be in the midst of a considerable increase in unreliability. The reason behind it is process, voltage, and temperature variations, semiconductor aging, and early life failures [1]-[3].

Amongst diverse dependable threats increased transistor aging because of Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI), can greatly affect the reliable operation of nano-scale circuits, designers have guaranteed reliability by various methods such as guard banding and conservative operating conditions [4],[5].Transient faults and soft errors do not last for a small period. Occasionally these errors do not damage a system, sometimes they can propagate and amass, and may cause system breakdown. In order to evade these different traditional styles with margining or error detection and correction, for instance Razor [6] and built-in soft error resilience (BISER) [7], can be applied against transient faults and soft errors, or by utilizing different conventional fault tolerance techniques based on redundancy. These techniques come at a high cost in area and power, threatening to a great extent to weaken the benefits of technology scaling [1].

The previously mentioned challenges can be dealt by leveraging main property of many in progress and growing application domains corresponding to multimedia (audio, video, and image) process, machine learning, data processing, search, and analytics—their computations may be executed approximately without significantly impacting the standard of results[8]-[10].This paper suggest that these applications may be designed with simplified error checkers that compute an approximation of the correct output, probably leading to a small likelihood of undiscovered faults, whereas still maintaining acceptable output quality.

This paper proposes an approach to design approximate error checkers using stochastic logic called StoCK (stochastic checkers), where the numbers can be represented as signal probabilities of pseudorandom bit streams, in stochastic computing (SC) [11], [12]. The major

advantage of stochastic computing (SC) is that different arithmetic operations can be put into practice in a highly power, area capable manner. Another fascinating property of SC is that the preciseness of the computation increases more and more as the computation proceeds. Thus, an approximation of the final output can be found from the initial bits of the output bit stream. Additionally, stochastic circuits square measure themselves quite fault tolerant, a number of bit flips in a bit stream do not impact the output significantly. These characteristics create SC promising for the designing of low overhead error checkers.

The stochastic error checker may lead to two major challenges. Firstly, the intrinsic approximate nature of stochastic computing (SC) can lead to outputs being wrongly identified as erroneous (false positive) or either some errors

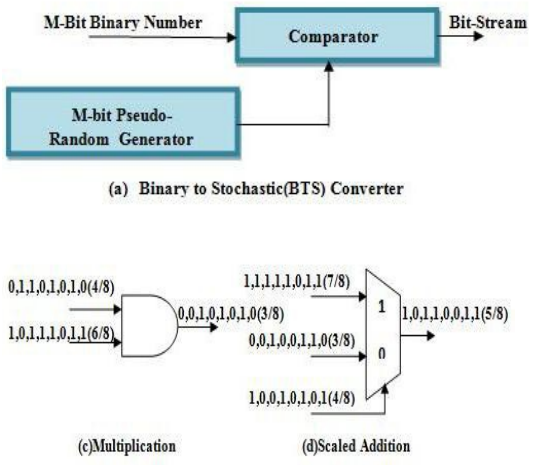

Fig.1.Stochastic circuit components

being undetected (missed coverage).Secondly, since stochastic circuits operate on bit streams, hence they may require longer time to complete, resulting to an error detection latency. Design techniques to optimize stochastic checkers for key metrics of, false positive, and detection latency is been proposed.

Stochastic computing (SC) has seen substantial interest in recent times, we are unaware of extra efforts to discover them as error checkers. Deliberate introduction of input-dependent errors lead to depravation in the algorithmic implementation, which is balanced by algorithmic noise-tolerance (ANT) schemes [13], the key distinction is that the use of stochastic logic for the design of error checkers, which brings unique benefits and challenges.

Section II provides the basics of stochastic computing (SC) and motivates its use for design of error checkers. Section III provides the complete description of the design of stochastic checker. In section IV,V simulation results and conclusion and future scope are given respectively.

PRELIMINARIES AND MOTIVATION

Above Fig.1 shows the stochastic circuit components. In stochastic computing [11],

data representation and process are fulfilled in the form of pseudorandom bit streams, so that the probability of 1in the set of bit stream corresponds to the magnitude of the number that is represented. Fig.1(a) and 1(b) shows the depiction of Binary- to- Stochastic(BTS) and Stochastic- to- Binary (STB) respectively. Further details of this components is given in [12].One of the main benefits of SC is that arithmetic operations, such as multiplication, addition, and others, can be realized in a extremely compact manner. as an instance, as depicted in Fig. 1c, a single AND gate can be used as multiplier. In the same way, a MUX performs scaled addition as in figure Fig. 1(d) of input bit streams. Thus it can be said that stochastic circuits has a likelihood to reduce error detection overheads.

Linear feedback shift registers (LFSR) are used to produce pseudorandom bit streams and are used in conversion of binary number to stochastic number. An M- bit polynomial LFSR can traverse through all the 2Mstatesexcept zero during a random sequence. Further the M-bit random number and the M-bit binary number are then compared, if the random number is less than or equal with binary input ″1″ is produced else ″0″. This make sure that at the end of 2M cycles number of ones in the final bit stream is equal to binary number. A counter is used in the STB, which accrue the 1s in the stochastic bit stream to give it‘s binary equivalent.

Stochastic circuits are highly efficient concerning to area and power, but one major disadvantage of it is high latency of circuit operation. Different techniques such as segmented stochastic representation, vector processing [13],[14] are given to perk up the performance while maintaining the compactness of SC. To overcome this drawback in this paper a different use of stochastic circuits as error checkers is explored, here speed will impact only error detection latency but not the performance of the circuit. The latency of the stochastic checker limits some types of faults from being detected like transient faults caused by soft errors and faults that are constant or changing slow due to ageing process, temperature, or intentionally caused due to voltage and clock over scaling can be identified with a notably lower overhead [15].SC engage in processing numbers encoded as probabilities that is represented in stochastic bit stream.

Let N1 signify number of 1‘s in a stochastic stream with Ns bits. It is

illustrated as N1/ Ns. The same is represented as (2 N1- Ns)/ Ns in bipolar. The stochastically encoded signal is precession restricted by the length of the sequence Ns. The minimum distance between representable number is [16]

$$\Delta_s = 1/N_s \quad \text{..... (1)}$$

Then quantization error is bounded by

$$-\frac{\Delta_s}{2} < e_{qs} \leq \frac{\Delta_s}{2} \quad \text{..... (2)}$$

The mean and variance of the stochastic quantization can be given as below respectively.

$$E(e_{qs}) = 0 \quad \text{.....(3)}$$

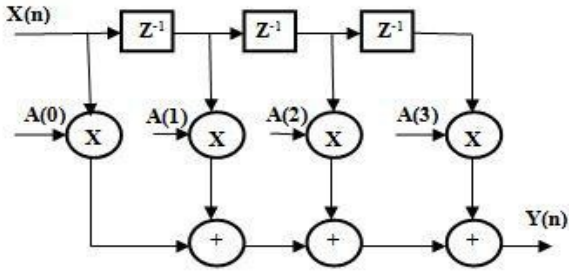

Fig.2. Four tap FIR filter

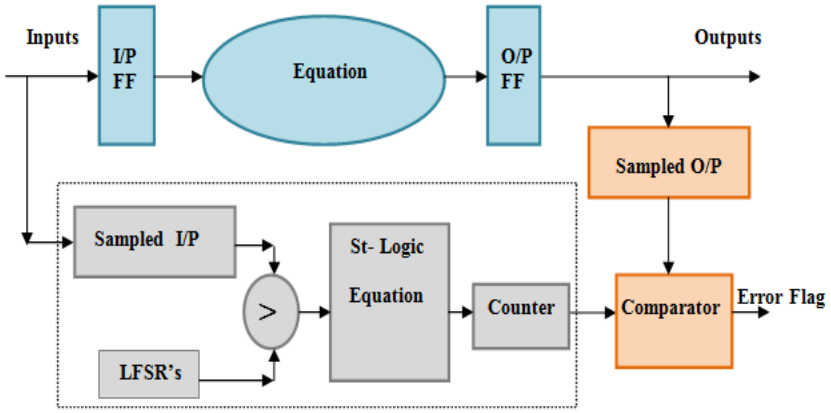

Fig.4. Stochastic checker for equation

$$\sigma_{qs}^2 = \frac{\Delta_s^2}{12} = \frac{1}{12.N_s^2} \quad \text{.....(4)}$$

As input bit streams are probabilistic in SC, the output of the stochastic circuit is inherently approximate. So, with a stochastic circuit used as a checker, so a small no of errors may be left out or some output may be falsely believed to be error.

Hence intrinsically error resilient applications are concentrated in this paper.

III PROPOSED WORK

Proposed stochastic logic is applied for Four tap FIR filter one of the most widely used circuits in digital systems and a general equation which are widely used. And further it‘s been applied to Triple modular redundancy(TMR) to depict the advantages of StoCK. This section explains the details of the design.

FIR filters are significant elements in Digital Signal Processor (DSP) owing to their linear phase frequency response. Fig.2 shows the four tap FIR filter its hardware implementation require multipliers, adders, and delay elements, which can be implemented as D flip flop. In an FIR filter X[n] is the filer input A[n-1] is the filter co-efficient the output Y[n] is produced over many clock cycles down the delay pipeline N is the order of the filter.

$$Y[n] = \sum_{i=0}^{N-1} X[n]A[n-1] \quad (5)$$

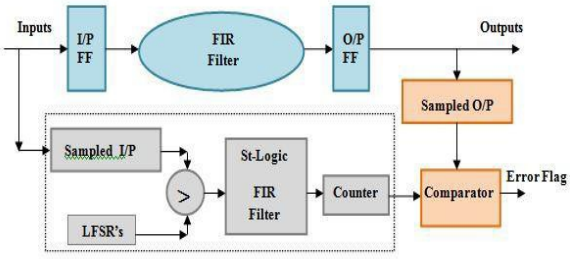

Fig.3. Stochastic checker for FIR filter.

Fig.3 depicts the block diagram of the stochastic checker for FIR filter. Here we have a stochastic circuit which execute the same function as the binary circuit which is implemented with necessary stochastic logic. Necessary BTS logic is used to convert the binary input to stochastic sequence as explained before and is given to the St-Logic FIR filter where the output is computed using stochastic input, whose output is converted back to binary using STB. As the latency of the original circuit and stochastic circuit vary the input and output of the original circuit are sampled. Once both the circuits performs the evaluation, the binary circuit and stochastic circuit output are compared to find out if there is an error, and error flag is raised.

B. Equation.

Equations are expressions which are commonly used in diverse applications, which involve variables and coefficients which can perform various mathematical operations such as multiplication, addition

and division. Two such operations can be altered, to one form to another by employing properties of associativity, commutativity, distributativity etc.

$$z=x1 + x2 + x4 + x3 +(1- x4) \quad (6)$$

Fig.4 depicts the block diagram of the stochastic checker for equation. Here we have a stochastic circuit which execute the same function as the binary circuit which is implemented with necessary stochastic logic.

Necessary BTS logic is used to convert the binary input to stochastic sequence as explained before and is given to the St-Logic FIR filter where the output is computed using stochastic input, whose output is converted back to binary using STB. As the latency of the original circuit and stochastic circuit vary the input and output of the original circuit are sampled. Once both the circuits performs the evaluation, the binary circuit and stochastic circuit output are compared to find out if there is an error, and error flag is raised.

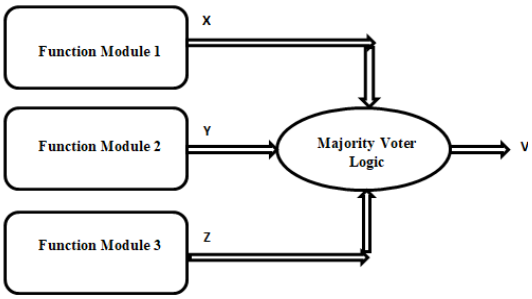

Fig .5. Block diagram of TMR.

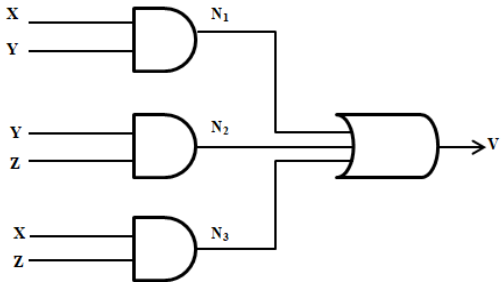

Fig.6. Majority Voter

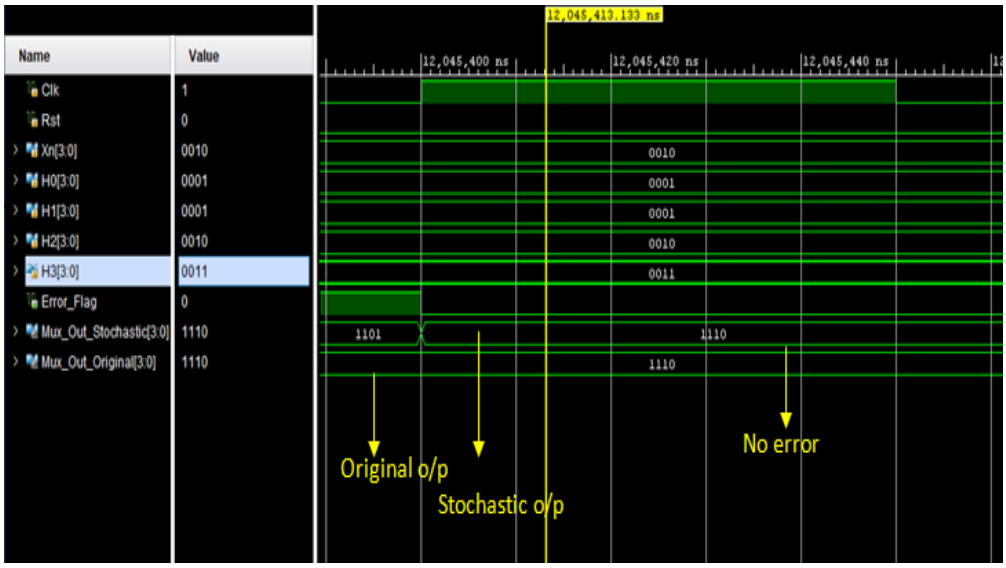

Fig.7.4 tap FIR filter simulation window.

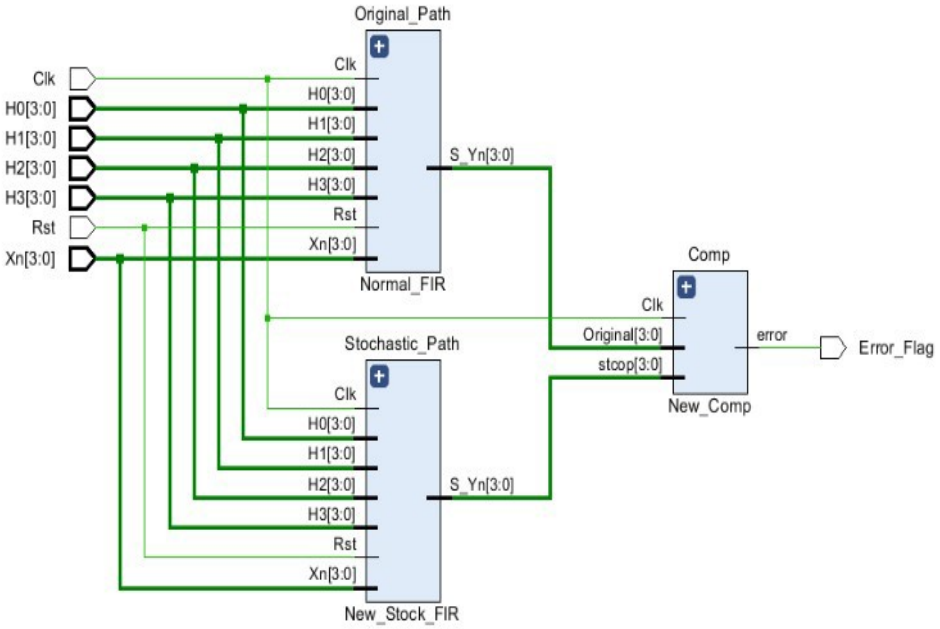

Fig.8.RTL Schematic of FIR filter

C. Triple Modular Redundancy (TMR)

TMR is a general method that can be applied to, sequential logic, Combinational logic, routing elements and memory cells independently or with grouping in a digital design.

In TMR often hardware overhead produced is more (about 200%). TMR, is a division of N-modular redundancy, it include two-times replication of a function module so that the three identical function modules are combined with a voting element as depicted in Fig.5.Where

function modules 2 and 3 are mainly copies of the function module 1. X, Y and Z denote the corresponding (equivalent) outputs of function modules 1, 2 and 3, which form the primary inputs to the voter, whose output is labeled as V. If any random function module becomes faulty, the TMR system would still carry on it‘s operation properly because of the Boolean majority, which is established by the voter through (7). In (7), sum means logical disjunction and product means logical conjunction.[17]

$$V = XYZ + XY + YZ + XZ = XY + YZ + XZ \quad (7)$$

The voter, depicted in Fig6, consists of three 2-input AND gates in the first level and a 3-input OR gate in the second level, which synthesizes (7). X, Y and Z correspond to the primary voter inputs, which signify the equivalent outputs of preceding and identical function modules. V represents the voter‘s output which synthesizes (7).

The labels N1, N2, N3 of the voter represent interconnects/internal output nodes of the first-level AND gates in Fig6. Ideally, the function modules outputs viz. X, Y and Z which serve as the primary voter inputs are either 0‘s or 1‘s, which are specified as the _no function module fault‘ Note that any other input combination of X, Y and Z signifies _single/multiple function module faults.

The above TMR is applied to both Four tap FIR filter and polynomials equation, where three copies of both modules are taken and the voter logic has been applied as explained above and has been simulated and verified for various values.

SIMUNATION RESULTS

A Verilog code has been written for four tap FIR filter, equation and an TMR as in Fig.3, Fig.4 and Fig.5. These designs are simulated and synthesized for different possible inputs.

In both Fig.7 and Fig.9 for StoCk checker, at reset equal to _0‘ pseudorandom sequence is generated and is compared with the same binary input given to original circuit and a stream of stochastic bits are generated through BTS, which is given to St-logic circuit and its output is computed and its converted back to binary number and both original binary circuit output and stochastic checker output is compared to give a‘0‘ if equal or _1‘ if different as error flag.

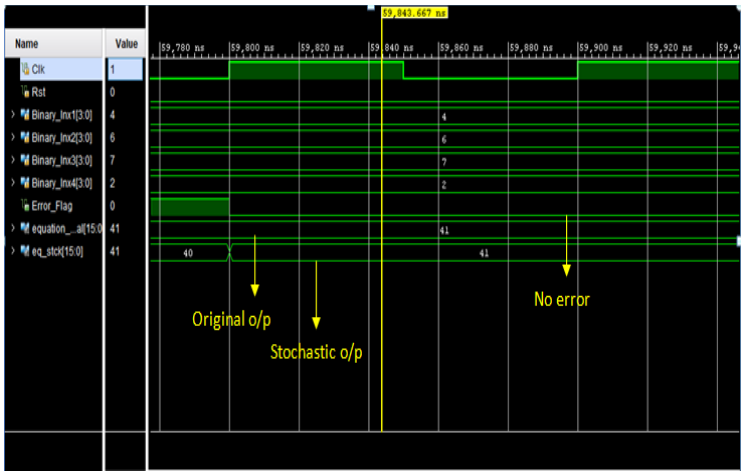

Fig.9. Equation simulation window

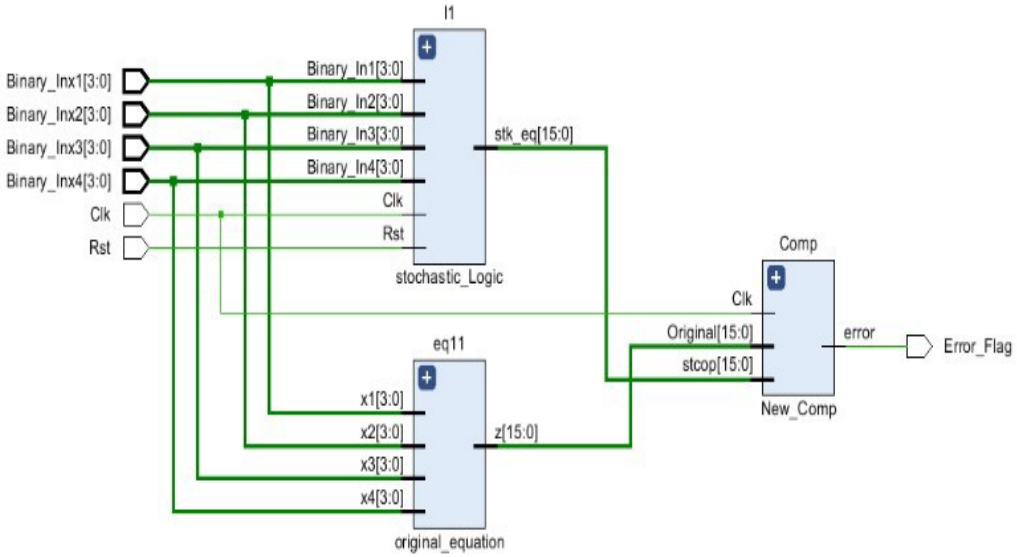

Fig.10.RTLSchematic of Equation

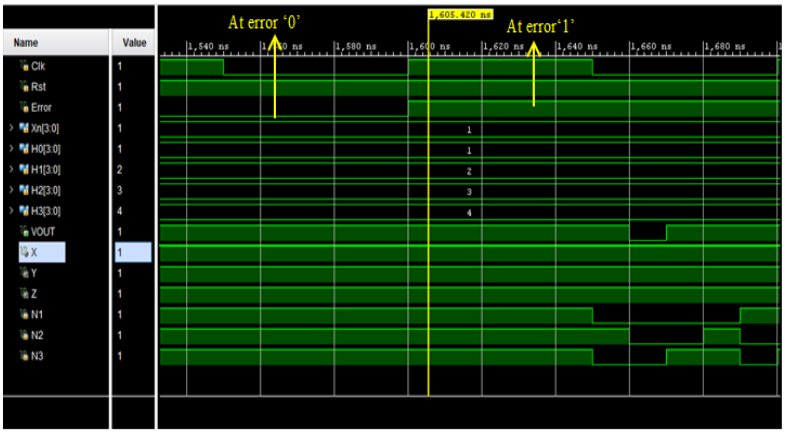

Fig.11.TMR simulation window for FIR filter

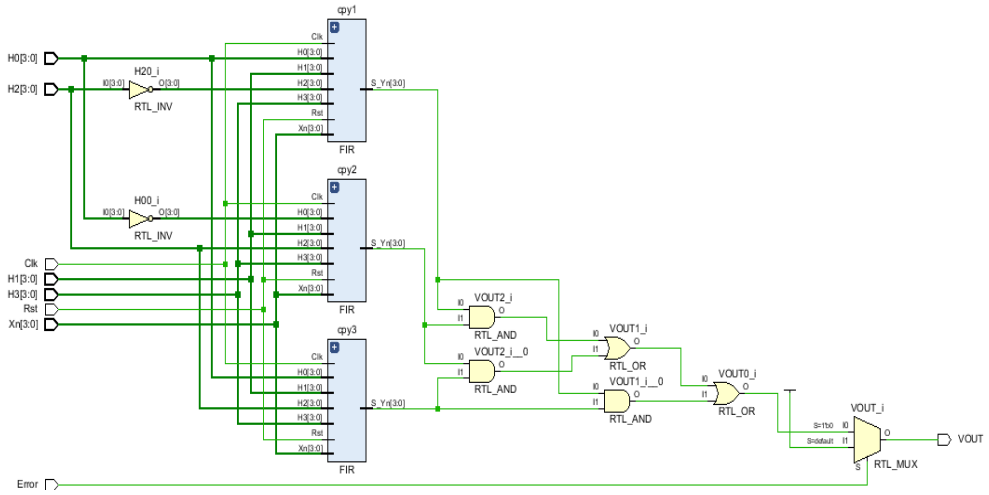

Fig.12.RTL Schematic of TMR for FIR filter

In Fig.7 Xn is given _0010‘,H[0] is given as _0001‘, H[1] is given as _0001‘, H[2] is given as _0010‘,and H[3] is given as _0011‘ and the original output is _1110‘ and at some clock cycle the stochastic logic circuit output is equal to original binary circuit output and error flag is _0‘.The RTL schematic of the four tap FIR filter is as shown in Fig 8.

In Fig.9 when Reset is equal to " 0" Binary-InX1 is given " 0100" , Binary-InX2 is given " 0110" Binary-InX3 is given " 0111" , and Binary-InX4 is given " 0010" and the original output is " 00101001" and at some clock cycle the stochastic logic circuit output is equal to original binary circuit output and error flag is " 0" .The RTL schematic of the equation is as shown in fig.10.

In fig.11 for TMR FIR filter when reset is equal to _1‘ and error flag is set to _0‘, then XYZ equal to _111‘ respectively vout‘1‘which shows no error, further when error flag is set to _1,‘an error is induced such as stuck at _1‘ or stuck at _0‘ through which vout is calculated respectively. The RTL schematic of the TMR FIR filter is as shown in fig.12.

In Fig.13, when three copies of equation circuit is considered with the voter logic and coded and simulated. when reset is equal to _1‘ and error flag is set to _0‘, then XYZ equal to _111‘ respectively vout‘1‘which shows no error or no functional module failure is present, further when error flag is set to _1,‘anerror is induced such as stuck at _1‘ or stuck at _0‘ through which vout is calculated respectively..The RTL schematic of the TMR equation is as shown in Fig.14.

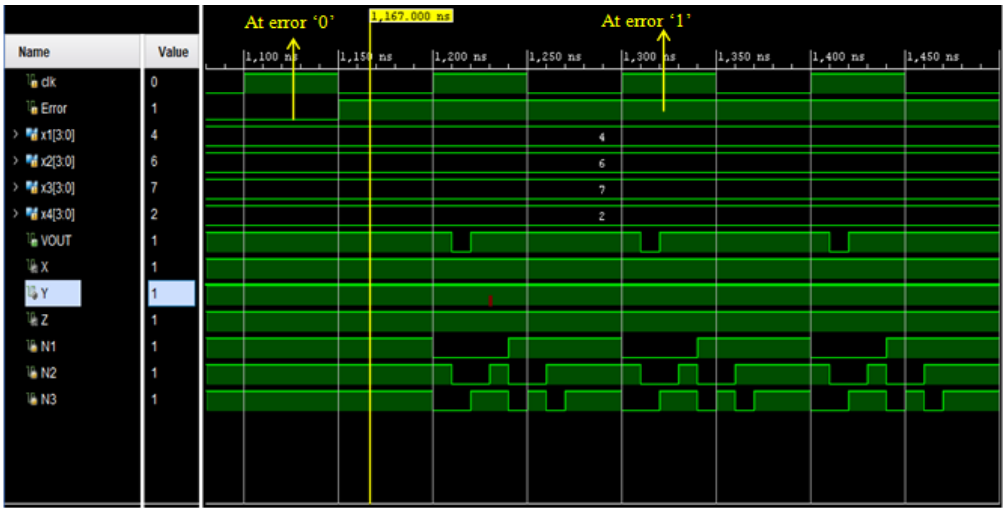

Fig.13.TMR simulation window for equation

CONCLUSION AND FUTURE WORK

There has been an increasing interest in stochastic computing (SC) recently. A different use for the stochastic circuits have been explored by designing stochastic checkers to check different digital circuits. Compared to Triple Modular Redundancy [TMR] the area overhead is reduced, as in TMR three copies of the circuit is used.Limitation of the proposed concept is that it fails to detect permanent faults caused by temperature variation and circuit aging.

A Four tap FIR filter is been simulated using Xilinx Vivado design suit and the on chip power consumed and a thermal margin is as show in Table1. Compared to Triple Modular Redundancy [TMR] the area overhead is reduced, as in TMR three copies of the circuit is replicated, where area overhead is increased and even the cost.

Equation is been simulated using Xilinx Vivado design suit and the on chip power consumed, and a thermal margin is as show listed in Table1. Compared to Triple Modular Redundancy[TMR] the area overhead is reduced, as in TMR three copies of the circuit is replicated, where area overhead is increased and even the cost. Further the Stochastic checker to be explored to various other applications. Implementation can be carried out on the FPGA kit.

TABLE 1 : Power and Thermal Margin

FIR filter	Equation
Power=1.644W	Power=1.805W
Thermal margin= 41˚C	Thermal margin =39. 2˚C

REFERENCES

I. J. Van Horn, ―Towards achieving relentless reliability gains in a servermarketplace of teraflops, laptops, kilowatts, and ―cost, cost, cost‖...:Making peace between a black art and the bottom line,‖ p. 8.inProc.ITC,2005,[Online].Available:http://ieeexplore.ieee.org/document/1584029/

II. The International Technology Roadmap for Semiconductors (ITRS), System Drivers,2013 [Online]. Available: http://www.itrs.net/

III. Rahimi, L. Benini, and R. K. Gupta, ―Hierarchically focused guardbanding: An adaptive approach to mitigate PVT variations and aging,‖ in Proc. Design, Autom. Test Eur. Conf. Exhibition, pp. 1695–1700, (DATE), Mar. 2013.

IV. M. Ebrahimi, F. Oboril, S. Kiamehr, and M. B. Tahoori, ―Agingaware logic synthesis,‖ in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), pp. 61–68, Nov. 2013

V. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull, and M. Zhang, S. Mitra, T. M. Mak, N. Seifert, N. J. Wang, Q. Shi, K. S. Kim, N. R. Shanbhag, and S. J.. (VLSI) Syst., vol. 14, no. 12, pp. 1368–1378, Dec. 2006.

VI. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

VII. N. Gala, V. R. Devanathan, K. Srinivasan, V. Visvanathan, and V. Kamakoti, ―ProCA: Progressive configuration aware design methodology for low power stochastic ASICs,‖ in Proc. 27th Int. Conf. VLSI Design, 13th Int. Conf. Embedded Syst. .pp. 342–347, Jan. 2014V.

VIII. K. Chippa, S. T. Chakradhar, K. Roy, and A. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC) , pp. 1–9, May 2013.

IX. B. R. Gaines, ―Stochastic computing,‖ in Proc. Spring Joint Comput. Conf. (AFIPS), pp. 149–156 New York, NY, USA, 1967.[Online].Available:http://doi.acm.org/10.1145/1465482.1465505

X. A. Alaghi and J. P. Syst., vol. 12, no. 2s, pp. 92:1–92:19,May2013.Online].Available: http://doi.acm.org/10.1145/2465787.2465794

XI. R. Hegde and N. R. Shanbhag, –Energy-efficient signal processing via algorithmic noise-tolerance,‖ in Proc. Int. Symp. Low Power Electron. Design, pp. 30–35, Aug. 1999.

XII. V. K. Chippa, S. Venkataramani, K. Roy, and A. Raghunathan, –StoRM: A Stochastic Recognition and Mining processor,‖ in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED) , pp. 39–44, Aug. 2014.

XIII. Neel Gala, SwagathVenkataramani, AnandRaghunathan,Fellow, IEEE, and V.Kamakoti, –Approximate error drtection with stochastic checkers‖ IEEE Transactions On Very Large Scale Integration Systems,2017.

XIV. Ran Wang ″ Design and Evaluation of Stochastic Circuits for FIR Filters and Vector Quantizers ″ A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science, Department of Electrical and Computer Engineering University of Alberta,2015.