IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 3, Issue 2 (Sep. – Oct. 2013), PP 11-16
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Fully Pipelined Power-Optimization Implementation of Monte Carlo Based SSTA on FPGA

S.V.Krishna Rao[1], Bura.Anusha[2]
[1](Assistant Professor, EIE, KITS, Warangal, India)
[2](VLSI&ES, KITS, Warangal, India)

Abstract: *Monte Carlo based SSTA serves as the golden standard against the alternative SSTA algorithms. The efficient implementation of MC-SSTA is performed by repeatedly executing ordinary STA using a set of randomly generated delay samples. The FPGA device is used as a target device onto which the RTL description is mapped, which acts as a dedicated STA engine. We leverage the path level and gate level parallelisms and the power optimizations of normal distribution random number generators based on central limit theorem. The accuracy is compared with the Mersenne Twister and Box Muller Methods which are high quality random number generators.*

Keywords**:** *FPGA, Linear feedback shift register, Monte Carlo, SSTA.*

I. Introduction

From the early times, accurate and efficient timing analysis is being demanded for a competitive design of a circuit. As the IC Technology scales down to nanometer range, the manufacture variations arise along with the tightened timing constraints. The allocation of active devices and their interconnections is a major constraint. The timing analysis has become a major issue in modern design environment; with the increased frequencies timing closures are to be reduced implicitly to meet the desired margins. Thus to ensue the desired goals one should concentrate on timing constraints, and also on the CMOS process variations.

Static timing analysis is one of the many available techniques to verify the timing of a design. The timing analysis simply refers to the analysis of design for timing issues. The STA is static as the analysis is carried out statically which refers that it is independent of the data values being applied at the input. The static timing analysis is contrast to the simulation based timing analysis where the verification is obtained through the applied stimulus.

The latest arrival times are propagated through all the logic gates and the critical paths are indentified. The computational complexity is O(N) when the no: of gates in a circuit is N, as the STA is based on simple graph tracing. To cope up with the process, voltage and temperature variations and to deal with the conservatism which is considered as the drawback of the STA; in a viewpoint, a worst case analysis is being traditionally employed [1], as the correlations of the gates are assumed the timing report tends to be pessimistic.

The Statistical STA has become a important tool to avoid the correlations and the parameter variability is dependent on timing analysis obtained from SSTA [2]-[4], which is statically more correct than the original STA. Major algorithms of SSTAs- Block based [3], [5]-[7] and Path based [8]-[10] approaches are presented. Path based technique evaluates the timing constraints of the paths between the blocks in a design in topological order. The computational complexities with respect to the traditional STAs [11] are almost the same. The timing distributions of all the paths are max-ed to virtual endpoint are obtained by adopting the canonical timing models.

Timing distribution using the Clark's approximation is presented [2], [5]-[7]. Normality assumption is adopted for long path analysis where there is a chance of occurrence of averaging effects, this assumption is rarely applicable for short path analysis. Both the paths are averaged to avoid the the hold error. The non-normal distributions are handled by timing model and timing analysis framework.

The promising approach for non-normal distributions is to apply a Monte Carlo method for timing analysis [12], [13]. Monte Carlo methods directly use delay samples which handles the arbitrary timing distribution model. The Monte Carlo analysis is a delay model independent, for which the accuracy increases with the delay samples, whereas the conventional STA repeatedly executes with the randomly generated delay instances to calculate timing distributions.

To overcome the computational intensiveness of the statically static timing analysis i.e., MCSSTA, the large no: of repetitions are required. The efficient approach which utilizes low discrepancy random numbers, Latin hypercube sampling are discussed [13], Non parametric max/min operations based on Mann-Whitney statistics is defined to propagate efficient timing vectors [12]. More distinguished approach is given in Monte Carlo based SSTA [14].

By solving the optimization problem, we can obtain the max speed up and execution throughput on limited FPGA area. The acceleration for each logic gate can be performed by a dedicated delay sample generator. By this idea an power optimized pipelined implementation of MCSSTA is proposed, in which a target circuit to be analyzed is translated to synthesizable RTL description, which is then mapped to a target device. The proposed flow and example of transformation of target net list into MC-SSTA is shown in fig 1.

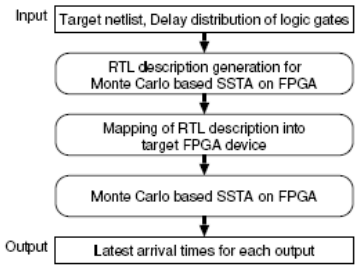

Fig.1. Proposed MC-SSTA flow

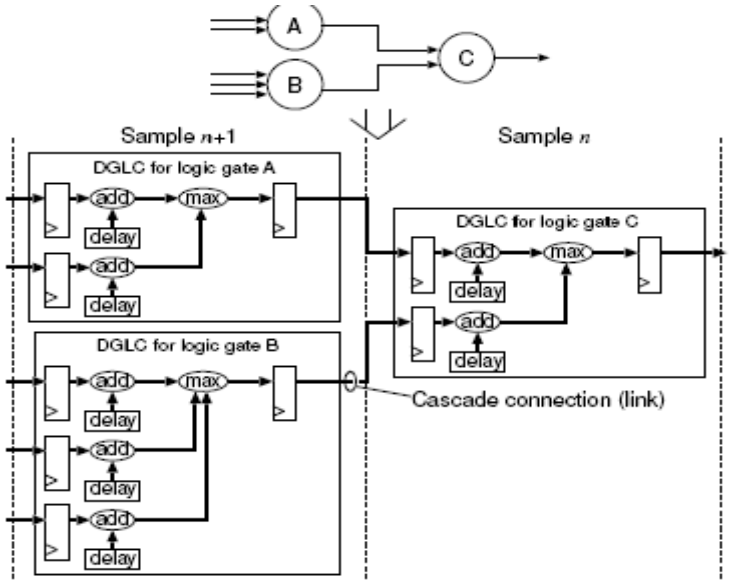

Fig. 2. Transformation of target net list into MC-SSTA implementation on FPGAs

The paper is organized as follows; a brief review of the translation procedure is presented in Sec. II. The details of proposed architecture in Sec. III. The Sec. IV discusses the results and we conclude the paper in Sec. V.

II. Overview Of Monte Carlo Based Ssta

In this section, we review the general procedures. A long path analysis that utilizes add-max as the basic STA is used as an example; the same discussions are equally valid for short path analysis with add-min operation. By repetitively running the STA on a target net list with various delay patterns a MC-SSTA is realized, the procedure includes:

1. A delay sample is generated for each delay arc in logic gate.
2. On a general purpose processor a STA is run on Monte Carlo sample.
3. The latest arrival times i.e., LAT are stored and the procedure is again repeated for the other cycle form the beginning.

With sufficient hardware support, each delay sample of a target circuit can be generated, which is suited for inherent parallelism for acceleration of hardware.

III. Implementation Of Fully Pipelined Monte Carlo Based Ssta

A. Design flow

The proposed flow of MC-SSTA is shown in fig.1, the gate delay distributions are assumed to be represented as the normal distributions, the mean and the standard deviation of each arc is given as floating number. A synthesizable RTL description is generated is generated from the given input data, then the description is mapped to a target device i.e., FPGA Spartan3E. To pursue parallelism each STA operation is realized by a logic gate, each and every gate in circuit is replaced by a DGLC- Delay sample generator and LAT time calculator. The delay samples of delay arc in the gate are generated and STA operation is executed. The DGLCs are connected cascaded, which forms as a link. Different circuit yields different construction of DGLCs, as the circuit changes then the implementation of hardware also changes.

B. Architecture of DGLC

A DGLC includes the normal distribution random number generator, linear feedback shift registers, adders and comparators. The comparator is used to decide max/min for long/short path analysis. The NDRNG-Normal Distribution Random Number Generator is used to obtain the normally distributed random numbers from the uniform numbers which are generated by a LFSR, which is a sub part of NDRNG. The fig.3, illustrates the construction of DGLC, each input corresponds to the adder and the NDRNG, the no: of adders and NDRNG is same. A four input DGLC is represented; as a result we have four adders and four NDRNGs.

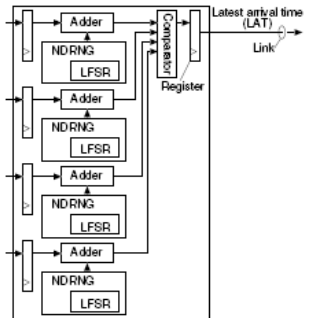

Fig.3. construction of DGLC (4 inputs)

LATs from preceding DGLCs and delay samples are added by adders to calculate output arrival times. The latest arrival times of the four is selected as the output LAT by comparator and then sent to DGLCs through links. The frequency range of target device ranges between 30MHz to GHz order. N_{fr} is the no: of bits for fractional part of LATs.

C. Normal Distribution Random Number Generator

The RNGs are crucial in MC-SSTA as it limits the speed of generation of gate delay samples. The central limit theorem is utilized as it includes simple arithmetic operations. The normal distribution random numbers from uniform random numbers Ϫi € (0, 1] are generated by

$$N(\mu,\sigma)\sim\sigma\times\frac{2\sqrt{3}}{\sqrt{N}}\left(\sum_{i=1}^{N} Xi-\frac{N}{2}\right)+\mu \qquad (1)$$

μ, σ and N are mean value, standard deviation and integer constant.

The NDRNG requires $N_{NDRNG}=(N\times N_{RNG})$ cycles to generate a normal distribution random number. Here N=12 is used to balance efficiency, accuracy and implementation.

D. Linear Feedback Shift Register

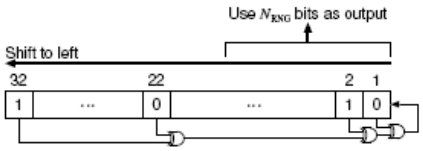

Fig.4. 32 bit LFSR

Considering the hardware cost including the Mersenne Twister MT [15], a liner feedback shift register is used in circuitry to reduce overall cost, A 32 bit LFSR is used by giving the randomly chosen initial stages to avoid unnecessary correlations. When N=12 in eq(1); we can avoid usage of same random numbers even for generating one million delay samples. The period of n-bit LFSR is 2^n-1, which is larger than one million

samples for n=32. The LFSR shifts by 1 bit every clock cycle and the tap locations where LSB is OR-Ed [16], are concentrated for the power optimization. The output bit is directly used as uniform random number.

E. Proposed Implementation

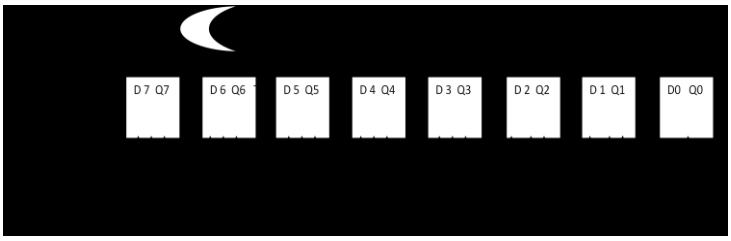

Fig.5. Proposed architecture of LFSR

Depending upon the logic used in feedback path, the register follows a predefined sequence, with a max seq length of 2^n-1 in an n-bit register. The default realization with a power-on initialization to the all zero states will not work, because the register could stay forever in all zero state, one obvious workaround is to use a power-on initialization to some other state or else change the feedback path.

IV. Simulation Results

The simulator results are generated by applying the clock signal and reset is enabled, we observe that the all-zero state is established, the reset is disabled and the simulator in invoked which distributes the seed value.

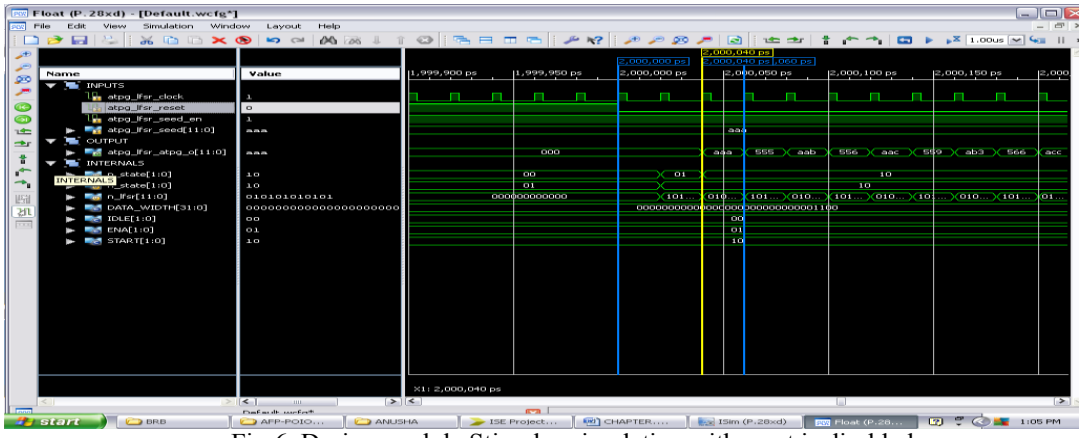

Fig.6. Design module Stimulus simulation with reset is disabled

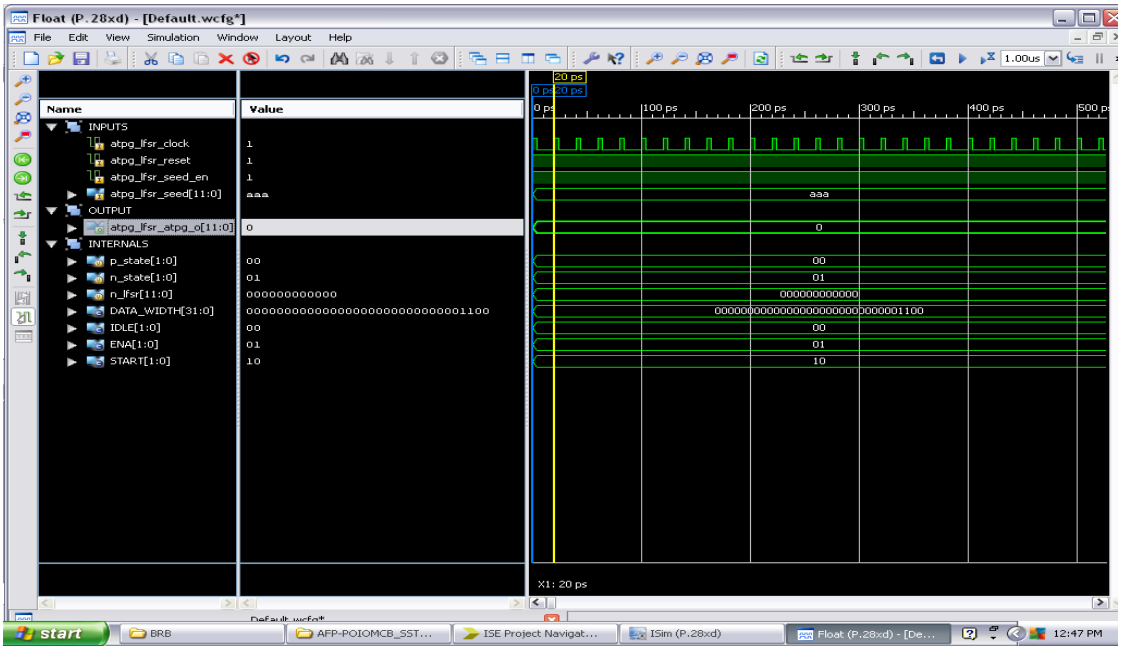

Fig.7. Design module Stimulus simulation with reset is enabled.

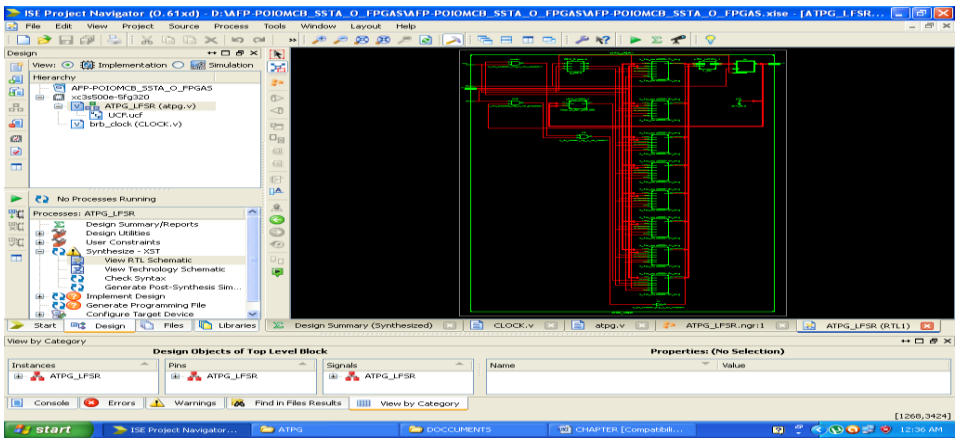

Fig.8. RTL Schematic Model of Internal logic

On-Chip Power Summary				
On-Chip	Power (mW)	Used	Available	Utilization (%)
Clocks	0.00	6	---	---
Logic	0.00	243	9312	3
Signals	0.00	436	---	---
IOs	0.00	29	232	13
BRAMs	0.00	2	20	10
Quiescent	80.98			
Total	80.98			

Fig.9. On Chip power summary and utilization

V. Conclusion

A Fully Pipelined-Power Optimization Implementation of Monte Carlo Based SSTA on FPGAS on semi custom Design, the main goal for designing it is to maintain speed, area and power which are successfully achieved. Power optimization: 50% duty cycle (FPGA) active clock cycle is power saved. The Spartan3E device is used with the operating frequency of 50MHz at a speed of -5. Further the scope is extended in terms of layout designing.

Acknowledgements

Siruvolu Venkata Krishna Rao , received his **A.M.I.E** in **E**lectronics and **C**ommunication **E**ngineering from the **I**nstitution **O**f **E**ngineers (India), Kolkata, India, in 2004 and the M.Tech degree in **VLSI** & **E**mbedded **S**ystems from Kakatiya University, Warangal, Andhra Pradesh, India in 2010. As per his interest in Personnel Management, He studied MIRPM (Industrial Relations and Personnel Management) from Alagappa University, Karaikudi, Tamil Nadu during 2004-2006. He worked for Indian Air Force in the Radar Division for considerable period. He is now working as Assistant Professor in the department of **E**lectronics and **I**nstrumentation Engineering in Kakatiya Institute of Technology and Science, Warangal, Andhra Pradesh, India since Jun 2008. His research interests include VLSI and Embedded Systems design, Networks on Chip Design for Testability. Mailing id :svkrishna_rao@yahoo.com.

Bura Anusha was born in Warangal, India in 1989. She received her B.Tech degree in **E**lectrical and **E**lectronics **E**ngineering from JNTU, Hyderabad in 2011. She is pursuing her Masters degree in VLSI & ES in Kakatiya Institute of Technology and Science, Warangal, India. Her research interests include Low Power VLSI Design and Digital Electronics. Mailing Id: anusha.bhura@gmail.com

References

[1] S. R. Nassif, A. J. Strojwas, and S. W. Director, "A methodology for worst-case analysis of integrated circuits," *IEEE Trans. CAD*, vol. 5,no. 1, pp. 104–113, Jan. 1986.

[2] Y. Zhang, A. J. Strojwas, M. Sharma, and D. Newmark, "Statistical critical path analysis considering correlations," in *Proc. ICCAD*, Nov.2005, pp. 699–704.

[3] X. Li, J. Le, M. Celik, and L. T. Pileggi, "Defining statistical sensitivity for timing optimization of logic circuits with large-scale process and environmental variations," in *Proc. ICCAD*, Nov. 2005, pp. 844–851.

[4] M. Pan, C. C.-N. Chu, and H. Zhou, "Timing yield estimation using statistical static timing analysis," in *Proc. ISCAS*, vol. 3, May 2005, pp.2461–2464.

[5] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *Proc. ICCAD*,Nov. 2003, pp. 621–625.

[6] J. Le, X. Li, and L. T. Pileggi, "STAC: statistical timing analysis with correlation," in *Proc. DAC*, Jun. 2004, pp. 343–348.

[7] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. DAC*, Jun. 2004, pp. 331–336.

[8] A. Agarwal, V. Zolotov, and D. T. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Trans. CAD*, vol. 22,no. 9, pp. 1243–1260, Sep. 2003.

[9] L. Lee, L.-C. Wang, T. M. Mak, and K.-T. Cheng, "A path-based methodology for post-silicon timing validation," *Proc. ICCAD*, pp.713–720, Nov. 2004.

[10] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *Proc. DAC*, Jun. 2004, pp.337–342.

[11] T. I. Kirkpatrick and N. R. Clark, "PERT as an aid to logic design,"*IBM J. Res. Dev.*, vol. 10, no. 2, pp. 135–141, Mar. 1966.

[12] M. Imai, T. Sato, N. Nakayama, and K. Masu, "Non-parametricstatistical static timing analysis: an SSTA framework for arbitrary distribution," in *Proc. DAC*, Jun. 2008, pp. 698–701.

[13] V. Veetil, D. Sylvester, and D. Blaauw, "Efficient Monte Carlo based incremental statistical static timing analysis," in *Proc. DAC*, Jun. 2008,pp. 676–681.

[14] J. Cong, K. Gururaj, W. Jiang, B. Liu, K. Minkovich, B. Yuan, and Y. Zou, "Accelerating Monte Carlo based SSTA using FPGA," in *Proc.FPGA*, Feb. 2010, pp. 111–114.

[15] M. Matsumoto, "Mersenne Twister home page." [Online]. Available:http://www.math.sci.hiroshima-u.ac.jp/_m-mat/MT/emt.html

[16] Hiroshi yuasa, Hiroshi Tsutsui, Hiroyuki Ochi and Takashi Sato, "A Fully Pipelined Implementation of Monte Carlo based SSTA on FPGAs" in 12 Int'l Symposium on quality electronic design, 2011.