

Standard Cell based Pseudo-Random Clock Generator for Statistical Random Sampling of Digital Signals

Rashed Zafar Bhatti

EE- Systems Department
University of Southern California
Marina del Rey, CA-90292, USA
bhatti@isi.edu

Keith M. Chugg

EE- Systems Department
University of Southern California
Los Angeles, CA 90089
chugg@usc.edu

Jeff Draper

Information Sciences Institute
University of Southern California
Marina del Rey, CA-90292, USA
draper@isi.edu

Abstract—A statistical random sampling technique has recently emerged as an elegant design-time efficient technique to address many timing issues of on-chip signals. To obtain reliable measurement results, it requires uniformly distributed sampling edges within the interval defined by the periodic cycles of the signal under measurement. This paper analyzes the characteristics of a random sampling clock relative to the signal under measurement and provides design rules for synthesis of pseudo-random sampling clocks. The proposed circuit provides a way to mix a range of frequencies in a pseudo-random fashion to produce uniformly distributed edges for random sampling, which yields measurement accuracy very close to that of true random sampling. A practical circuit design technique of pseudo-random clock generation is proposed based on a standard cell approach. This allows on-chip integration of a random clock generator while keeping the overall system design-time efficient and portable across varying technologies. The proposed designs are targeted to IBM Cu-08 standard cell libraries and provide a measurement resolution of 1ps.

I. INTRODUCTION

Timing uncertainty issues with control signal pulse width, duty cycles of clocks, relative phase of multiple clocks, clock skew with respect to data or strobe and relative skew in parallel data lines are classical digital design problems of synchronous systems. Rapidly scaling technologies and ever increasing interconnect delays exacerbate timing uncertainties for on-chip signals considerably. The commercial effect of rapid scaling reduces “Time to Market” (TTM) for consumer products, which leaves little margin for designers to perform tedious and time-consuming design and characterization iterations for custom components. To deal with typical timing issues, a statistical random sampling technique [3] has been proposed for observing and adjusting the timing of on-chip digital VLSI signals. This technique enables standard cell based designs which are inherently design-time efficient and portable across many VLSI technologies. The ability to provide a “random clock” is a crucial factor for applying this technique to accurately measure the timing of on-chip high-

speed digital signals. Theoretically the sampling clock required for the measurement through this technique should manifest a uniform distribution of its sampling edges in the time interval defined by the periodic cycle of the signal being measured, so that all parts of the signal under measurement are observable with equal probability. A practical realization of a truly random signal for statistical random sampling based measurement using a purely standard cell technology platform is not straightforward. This paper explores some design possibilities to generate an approximate random clock and their characterization relative to a range of signals expected to be measured. The rest of the paper is organized as follows. Section II presents related work and background. Section III discusses random clock generation and its analytical characterization relative to the signal under measurement. The proposed random sampling circuit is described in Section IV. Section V describes the test environment and empirical results, and section VI concludes the paper.

II. CONTEXTUAL BACKGROUND

The concept of statistical random sampling [1][2][3] as applied to digital VLSI signals can be understood from Fig. 1. The state of a periodic signal under measurement is repeatedly captured at random instants of time, which produces a sequence of independent and identically distributed (*iid*) Bernoulli random variables with probability of getting one equal to t_w/T in each trial. A large sample of this information is gathered, and the ratio of the captured number of ones in the sample converges to t_w/T . A prior paper [3] shows that if the sample size is carefully selected, then this parameter can be measured with very high accuracy and confidence level[8]. It is obvious that the success of the technique heavily relies upon the true randomness of the events at which the signal is observed. The rest of this paper

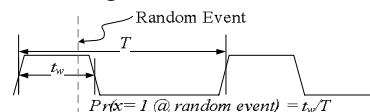


Figure 1. A periodic signal under observation.

deals with the issue of randomness related to these types of measurements.

III. CHARACTERIZATION OF A RANDOM CLOCK

Due to technological and TTM constraints, pure digital platforms do not allow the luxury of using sophisticated chaotic oscillators[13][14], PLLs [15] or programmable analogue delay components in design. Fig. 2 shows a conceptual block diagram of a pseudo-random clock generator employed by [3][4] to demonstrate the practicability of the random sampling technique for digital VLSI signals. A pseudo-random sequence of frequencies is generated by switching the control input of a ring oscillator using a control sequence of pseudo-random numbers. To characterize the random edges produced by this kind of pseudo-random clock, we first analyze the distribution of inter-arrival times of its sampling edges (generally the positive or negative edges). The mean of inter-arrival times of these edges corresponds to the average time period τ_{avg} and average frequency f_{avg} of these sampling clocks. It is very important to establish some bounds for this parameter.

A. Upper and Lower Bounds of Average Frequency

The lower bound of f_{avg} is not as critical as the upper bound, because it merely affects the measurement speed, i.e., a lower value of f_{avg} would require more time to gather a sample of a given size. Before establishing the upper bound for f_{avg} , it is noted that the random sampling technique finds its real worth for applications where the signal under measurement has so high a frequency that sampling it at higher frequencies may not be practical or possible. For the sake of completeness an upper bound for f_{avg} is set in terms of the minimum frequency $f_{sig-min}$ of the signal to be observed and maximum tolerable percentage error level ξ_{max} with a minimum desired confidence CI_{min} . The maximum average frequency of the random clock $f_{avg-max}$ is given by the following formula:

$$f_{avg-max} = f_{sig-min} (\xi_{max} n_{min}) \quad (1)$$

Where n_{min} is the size of the sample required for measurement with ξ_{max} and CI_{min} .

B. Distribution of the Edges of the Sampling Clock

To characterize the relation of the random clock with the signal under measurement, the arrival time of its edges along the periodic cycle of the signal must be analyzed. First a hypothetical ideal random clock is considered to establish a formulation. Then before going into the analysis of the

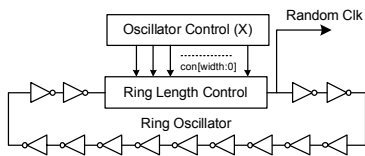


Figure 2. Block Diagram of the Controlled Ring Oscillators

frequency mix, which is blended using a sequence of pseudo-random numbers, the relation of a fixed-frequency sampling clock is analyzed with respect to a given signal under measurement.

1) Ideal Random Sampling Clock

When a particular signal is observed through statistical random sampling, its periodicity determines how an arbitrary distribution of the random edges would fold within its periodic interval. To understand this characteristic, consider a signal of time period τ_{sig} , sampled using a hypothetical ideal random clock. The inter-arrival times of its positive edges follow a true random sequence represented by $\{\tau_{x0}, \tau_{x1}, \tau_{x2}, \dots\}$. The following mathematical formulation determines how the random edges of this ideal random clock would fold within the periodic interval τ_{sig} of the signal under measurement.

$$t_{x0} = \varphi, \quad t_{x1} = (\tau_{x0} + t_{x0}) \bmod \tau_{sig}, \quad \dots, \\ t_{xi} = (\tau_{xi-1} + t_{xi-1}) \bmod \tau_{sig} \rightarrow 0 \leq t_{xi} \leq \tau_{sig} \quad (2)$$

Here φ is an arbitrary initial phase between the signal and the random clock. The time of occurrence of edges within the periodic interval τ_{sig} is given by $t_{xi} = \{t_{x0}, t_{x1}, t_{x2}, \dots\}$. It is obvious from (4) that if $\{\tau_{x0}, \tau_{x1}, \tau_{x2}, \dots\}$ is a sequence of uniform random numbers within any arbitrary interval then $\{t_{x0}, t_{x1}, t_{x2}, \dots\}$ would likewise be within 0 to τ_{sig} .

2) Fixed-Frequency Sampling

When sampled with a fixed-frequency sampling clock, the sampling instants of time within the periodic cycle of the sampled signal can be calculated with modulo τ_{sig} of the absolute times of occurrence of the edges. The following mathematical expression captures this fact:

$$t_i = (i \times \tau_{sclk} + \varphi) \bmod \tau_{sig} \rightarrow 0 \leq t_i \leq \tau_{sig} \quad (3)$$

Where τ_{sclk} is the time period of the fixed-frequency sampling clock and φ is the arbitrary initial phase between the signal and sampling clock. For a given combination of τ_{sclk} and τ_{sig} the linear congruence (3) produces a finite set of a number sequence $\{t_0, t_1, t_2, \dots, t_n\}$ called a group, where each member of the group is a discrete sample point within the interval 0 to τ_{sig} .

3) Sample Point Resolution

If the sample points are sorted by their absolute values, they form a set of equally spaced uniformly distributed

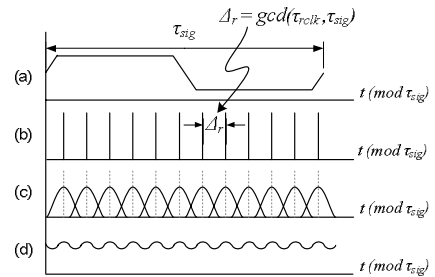


Figure 3. Fix frequency sampling of a signal.

discrete instants of time within τ_{sig} interval for a fixed frequency sampling clock as shown in the Fig. 3(b). The spacing between any two consecutive sample points on the time axis is defined as *sample point resolution* (Δ_r). It determines the maximum possible resolution with which a given fixed-frequency sampling clock can observe the signal under measurement. If the values of τ_{sclk} and τ_{sig} are represented by pure integers with equalized exponents, then the value of Δ_r can be calculated by the greatest common divisor (gcd) [7] of τ_{sig} and τ_{sclk} .

$$\Delta_r = gcd(\tau_{sclk}, \tau_{sig}) \quad (4)$$

In the best case $\Delta_r = 1$, for a relatively prime combination of τ_{sig} and τ_{sclk} , and it produces the longest sequences of sampling points. A catastrophic case is when τ_{sig} exactly divides τ_{sclk} and there is only one sample point.

4) Pseudo Random Clock Produced by Frequency Mix

A pseudo random clock can be generated when the length of a ring oscillator is continuously changed using a given pseudo random sequence of numbers $\{\psi_0, \psi_1, \psi_2, \dots, \psi_m\}$ at its control port. If the time periods of the frequencies generated corresponding to this sequence are given by $\{\tau(\psi_0), \tau(\psi_1), \tau(\psi_2), \dots, \tau(\psi_m)\}$ then edges produced by the frequency mix produced by this circuit arrangement within the τ_{sig} interval can theoretically be represented by the following recurrence:

$$t_{x0} = \varphi, \quad t_{x1} = (\tau(\psi_0) + t_{x0}) \bmod \tau_{sig}, \quad \dots, \\ t_{xi} = (\tau(\psi_{i-1}) + t_{xi-1}) \bmod \tau_{sig} \rightarrow 0 \leq t_{xi} \leq \tau_{sig} \quad (5)$$

The above recurrence (5) is identical to that of an ideal random clock given by (2), except that the true random intervals of inter-arrival time of the random edges are replaced by a pseudo-random sequence of intervals. In this work we compare three types of switching sequences: (1) linear sequence, (2) LFSR based random number [9] and (3) cellular automata based random number sequences[11].

5) Effect of Jitter

For the sake of simplicity, the above analysis ignored the expected jitter in the signal generated by the ring oscillator. The jitter is a true random variable generally modeled as a zero mean Gaussian[5]. Fig. 3(c) and 3(d) show the effect of Gaussian jitter and the resultant distribution of edges of a fixed frequency sampling clock. Similarly, in practice, distribution of the sampling edges of a frequency mix

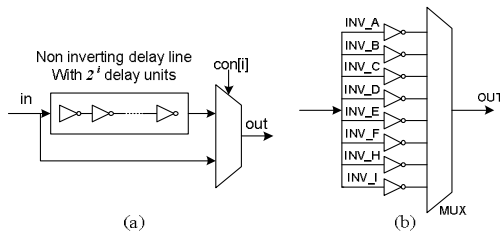


Figure 4. Components of the proposed PRCG
(a) Ring slice, (b) Delay blender

produced using a pseudo random number sequence appears to be true random because of the true random nature of the jitter effect. This randomness makes every measurement made using this frequency mix completely independent of previous ones.

IV. PROPOSED CIRCUIT DESIGN

A. Pseudo-Random Clock Generator (PRCG)

The proposed PRCG is a circular ring oscillator built with a series of non-uniform ring slices. Fig. 4(a) shows a ring slice. The delay-line of the ring slice, connected to the n^{th} bit of the control port, contains $N_{inv} = 2^{(n+1)}$ inverters. To produce a good frequency mix, the proposed PRCG design adds a delay blender in the ring oscillator. Fig. 4(b) shows a delay blender built with inverters of various performance levels. The delay blender enables this circuit to produce frequencies of very close values that increase the probability of getting frequencies of better Δ_r characteristics for a given signal periodicity. Most common commercial standard cell libraries provide a reasonably wide spectrum of buffers and inverters of different performance levels.

B. Random Sampling Unit (RSU)

Keeping in view the importance of Δ_r characteristics, the proposed circuit shown in Fig. 5 incorporates an additional counter, “Counter 0”. This counter is used to observe the relative characteristic of a random clock and the measured signal. This allows the controller to select a set of good frequencies and isolate bad ones relative to the signal under observation. At the start of a sample observation counter 0 is loaded with a value “S”. At run time counter 0 is decremented at every edge of the signal under observation. When it reaches zero, the value of “Counter 1” is captured in register “Q”. Δ_r is calculated from the following formula, a smaller value of Δ_r is indicative of higher measurement resolution.

$$\Delta_r = gcd(S \tau_{sig} / (n - Q), \tau_{sig}) \quad (6)$$

V. EXPERIMENTAL SETUP AND RESULTS

To validate the analytical models established in this work the proposed circuits were synthesized and targeted to IBM Cu-08 (90 nm) standard cell technology. A PRCG with 8-bit control width was implemented with the upper 5 bits

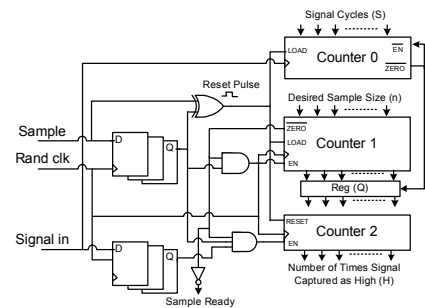


Figure 5. Proposed Random Sampling Unit (RSU).

connected to ring slices for coarse variation and the 3 lower bits connected to a delay-blender for finer variation of frequencies. The $f_{avg-max}$ of the frequency mix was set approximately to 100 MHz. To switch the control, three types of number sequences were evaluated: (1) linear sequence, (2) LFSR based random number [9] and (3) cellular automata based random number sequences[11]. The sampling distribution characteristics of pseudo random clocks were studied for 1GHz, 622.08MHz, 500MHz, 300MHz and 250MHz signals. Extensive simulations proved that our analytical models very closely track the actual behavior of the proposed PRCG circuit. The performance of three types of sampling clocks is compared on the bases of the resulting probability distribution of sample points in the τ_{sig} interval and the value of effective sample point resolution Δ_r -effective. Fig. 6 shows the Δ_r distribution of the constituent frequencies of the PRCG under test with respect to signals under observation. Fig. 7 shows the standard deviation of the sample point probabilities as an indication of smoothness of the approximately uniform distribution of edges produced by the random number sequence. The smaller values on this graph for linear sequence indicate smoother output distributions; both pseudo random sequences produced less smooth distributions. But pseudo random sequences produced Δ_r -effective = 1ps relative to different values of τ_{sig} , whereas the linear sequence produced Δ_r -effective = 2ps. Comparing the circuit complexities of LFSR and Cellular automata based random number generator, LFSR is better because of its simple and robust design. The measurement results of the RSU manifested a very high accuracy and close consonance with the expected theoretically estimated results. Our implementation of PRCG occupies 1120 cell units (cell = 0.28 μm x 3.36 μm for IBM Cu-08) and operates at 115.7384 μW which is an order of magnitude less than that of conventional analog frequency synthesizers.

VI. CONCLUSION

This paper analyzed the required characteristics of a pseudo-random clock for statistical random sampling and demonstrated a way to use well-known pseudo random number sequences to generate random instants of time. This work concludes that a well-crafted pseudo random clock relative to the signal under measurement can produce measurement accuracy very close to that of ideal theoretical random sampling. This makes the statistical random sampling technique an extremely practical and reliable solution to address many timing issues related to on-chip digital signals. The design rules provided in this work can be used to build a purely standard cell based PRCG which allow on-chip integration of a PRCG with a RSU while keeping the overall system area, power and design-time efficient and portable across varying technologies.

REFERENCES

- [1] R. Bhatti, M. Denneau, J. Draper, "2 Gbps SerDes Design Based on IBM Cu-11 (130nm) Standard Cell Technology", Proceedings of the ACM Great Lakes Symposium on VLSI, April 2006.
- [2] R. Bhatti, M. Denneau, J. Draper "Duty Cycle Measurement and Correction Using a Random Sampling Technique", Proceedings of the 48th IEEE International Midwest Symposium on Circuits and Systems, August 2005.
- [3] R. Bhatti, M. Denneau, J. Draper "Phase Measurement and Adjustment of Digital Signals Using Random Sampling Technique", Proceedings of the IEEE International Symposium on Circuits and Systems, May 2006.
- [4] S. Maggioni, A. Veggetti, A. Bogliolo, L. Croce "Random sampling for on-chip characterization of standard-cell propagation delay", Fourth International Symposium on Quality Electronic Design 2003.
- [5] W.J. Dally and J.W. Poulton, "Digital Systems Engineering", Cambridge University Press, 1998.
- [6] A. Leon-Garcia, "Probability and Random Processes for Electrical Engineering", Second Edition. published by Addison Wesley, 1994.
- [7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, "Introduction to Algorithms" (Second Edition) by published by MIT Press and McGraw-Hill.
- [8] Spiegel, Schiller, Srinivasan, "Theory and Problems of Probability and Statistics"; 2nd Edition, McGraw Hill.
- [9] G. Marsaglia, "A current view of random numbers," Computer Science and Statistics: The Interface, L. Billard, ed., Elsevier Science Publishers B. V., pp. 3–10, 1985.
- [10] D. E. Knuth, The Art of Computer Programming: Volume 2, Seminumerical Algorithms, 3rd ed., Addison-Wesley, ch. 3, 1998.
- [11] Shackleford, B.; Tanaka, M.; Carter, R.J.; Snider, G. "High-performance cellular automata random number generators for embedded probabilistic computing systems", Proceedings. NASA/DoD Conference on Evolvable Hardware, 2002
- [12] A. J. Johansson, H. Floberg, "Random number generation by chaotic double scroll oscillator on chip; IEEE International Symposium on Circuits and Systems, 1999.
- [13] G. Chen and T. Ueta, "Chaos in Circuits and Systems", ed 2002 (Singapore: World Scientific)
- [14] Fengling Han, Xinghuo Yu..., "n-scroll chaotic oscillators by second-order systems and double-hysteresis blocks", Nov. 2003
- [15] B. Vizvari, G. Kolumban, "Quality evaluation of random numbers generated by chaotic sampling phase-locked loops" IEEE Transactions on Circuits and Systems, Volume: 45 , Issue: 3 , March 1998.

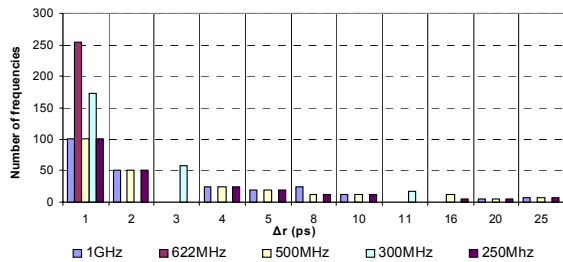


Figure 6. Relative Δ_r Distribution of the PRCG under test.

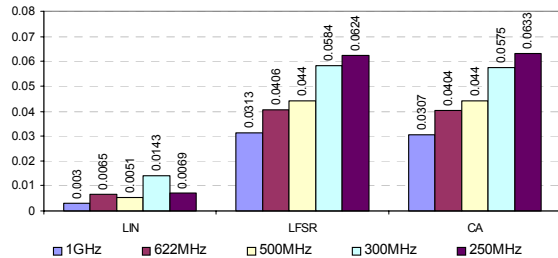


Figure 7. Standard Deviation of Sample Point Probabilities.