Stochastic Analysis of Bubble Razor

Guowei Zhang
Department of Microelectronics and Nanoelectronics
Tsinghua National Laboratory for Information Science and Technology
Tsinghua University
Beijing, China
fabregaszgw@sina.com

Peter A. Beerel[1]
Ming Hsieh Electrical Engineering Department
University of Southern California
Los Angeles, California, USA
pabeerel@usc.edu

***Abstract*—Bubble Razor has been proposed to eliminate required timing margins in synchronous design caused by increasing delay variation due to process variation and aging. However, the theoretical analysis of its performance under variability is unknown. This paper presents a Markov Chain model to describe the behavior of Bubble Razor. Using this model, we analyze its performance and provide an optimizing strategy to maximize its benefits.**

Keywords—Resilient design, variability, performance analysis.

I. Introduction

Traditional synchronous design must incorporate timing margin to ensure the correct operation under worst case conditions. However, the progressive increase in process variation and aging effect is causing increasingly large delay variations, requiring higher and higher timing margins, making traditional design less performance and energy efficient.

To address this problem, many techniques for resilient designs have been explored that address delay variations. Canary circuits [1] [2] have been proposed that predict errors employing a mimic delay chain that is configured during test. Designs can then adjust supply voltage or clock frequency either statically or dynamically to ensure the circuit is working at the edge of failure. In addition, Razor circuits [3] [4] [5] [6] have been proposed that contain in situ timing violation detection mechanisms that allows the circuits to recover from timing errors via architectural replay.

One challenge with these schemes is that legacy designs must be adjusted to analyze inserted error signals and implement replay, making adoption challenging. Bubble Razor was proposed to address this problem [7] [8]. Compared to previous designs, Bubble Razor uses a latch-based structure and automatically-generated clock gating logic that stalls the pipeline upon timing errors. In particular, upon a timing error, a Bubble-Razor design stalls its right neighbor locally and propagates the stalling signals (bubbles) from this stalled stage gradually throughout the entire design. This enables the design to have a maximum of one clock cycle penalty upon error and makes the scheme architecturally independent, enabling automatic application of Razor with no changes to existing RTL.

One issue with Razor circuits, including Bubble Razor, is that traditional clock-cycle-time based performance analysis cannot be used because the timing cost for correcting errors must be accounted for. In fact, to the best of our knowledge, no analytical analysis of Bubble Razor's performance under variability has been explored.

To address this open problem, we propose a Markov Chain model to describe the behavior of Bubble Razor. We analyze an N-stage pipeline ring and, based on our experiments, propose an approximate model of performance that can be used for more complicated circuits. We develop the analytical relationship between delay variation and *effective clock cycle time*, the average time to process each instruction. Using this model, we compare the performance of Bubble Razor with traditional synchronous circuits. Moreover, we use this approach to propose a method to maximize the performance of Bubble Razor circuits.

The paper is organized as follows. In Section II, we describe the behavior of Bubble Razor's error detection, correction, and propagation. Section III contains the detailed introduction to our Markov Chain model and the link between the model and circuit performance. In Section IV, we analyze the results of the model and present the comparison between Bubble Razor and traditional circuits as well as our approach for optimizing performance.

II. Bubble Razor

Bubble Razor (BR) inherits the features of previous Razor techniques enabling real-time error detection and correction [7] [8]. Unlike other Razor architecture, its novel latch-based bubble propagation algorithm makes it architecturally independent and enables the automatic application of this technique to legacy RTL designs, significantly reducing barriers to adoption.

Similar to other Razor architectures, Bubble Razor flags a timing violation when the data arriving at a latch varies after the latch opens. This is implemented by adding a shadow latch to record the data before main latch opens, and a corresponding XOR gate to each latch in the circuit. Upon detecting a timing violation, the circuit automatically recovers by stalling the subsequent latch giving it an additional clock cycle to process the data. Half of the additional clock cycle is used to compensate for the unexpectedly large delay from the previous latch and the other half accounts for the delay from the current latch to subsequent one. Thus after the stalling, the timing

[1] Peter A. Beerel also works as Chief Scientist in the Network Silicon Group at Intel Corporation.

978-3-9815370-2-4/DATE14/©2014 EDAA

violation has been corrected as long as the real delay of each half clock-cycle *step* never exceeds one clock cycle time.

However to ensure correct operation, stalling the subsequent latch is not sufficient. In fact, up-stream stages must be stalled to ensure valid data is not over-run and down-stream stages must be stalled to ensure corrupt data is not accidently interpreted as valid. Previous Razor structures use counter-flow pipelining or architectural replay to recover from the stall [3] [6], both of which techniques require the RTL to be designed with Razor in mind. However, the latch-based scheme in BR enables an automatic local stall propagation algorithm.

Consider the 2-stage ring (consisting of 4 latches) in Figure 1. A timing violation causes an error signal to be sent to its Right Neighbor (RN) to tell it to stall. Then, the stalling spreads both forward and backward directions around the ring in a wave-like pattern. The spreading of stalls is terminated by the stage which receives stalls from both directions in what is called *stall annihilation*.

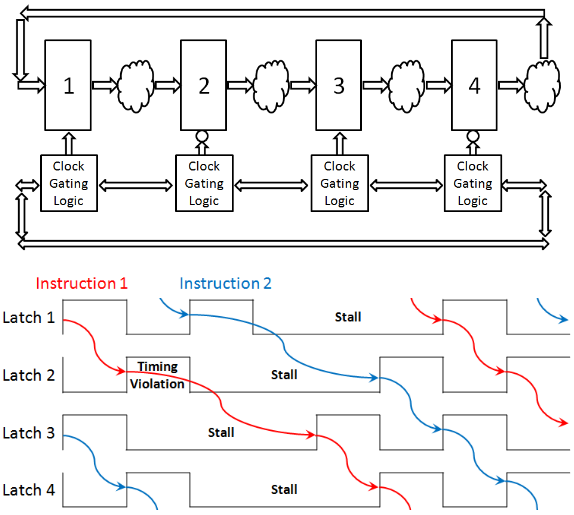

Figure 1. Bubble Razor block and timing diagrams.

This wave-pattern of stalling upon detecting an error is based on a bubble propagation algorithm. A latch that receives bubbles from some of its neighbors stalls and sends bubbles to its other neighbors. A latch that receives bubbles from all of its neighbors stalls but doesn't send out bubbles (i.e., *annihilation*). This bubble propagation algorithm guarantees the correctness of a BR design and is therefore the basis of our Markov Chain Model.

III. Markov Chain Model

Since the timing cost for correcting errors has to be considered, modeling the performance of a BR design based solely on the clock-cycle time is not sufficient. In a BR circuit, the state of individual latches, i.e. working or stalling, and the information of timing violations and bubbles define the state of the entire circuit. The circuit state is a stochastic variable which changes over time. Because the state of the latch in the next clock cycle is determined solely by its current state and not influenced by earlier conditions, the stochastic process is a Markov chain. Consequently, using Markov Chain analysis we can describe Bubble Razor's behavior and present a mathematical treatment of its performance.

A. Markov Chain Model for a Ring

We first consider an N-stage ring containing 2N latches with no primary inputs or outputs. There are two categories of states for a latch: working and stalling. Working means in this clock cycle the latch closes and opens normally while stalling implies that the latch does not open so that it does not accept new data from its input and keeps the output fixed in this clock cycle.

Moreover there are two possible working states: W (Working without timing violation) and E (working while a timing violation happens, E is short for Error). And in neither states W nor E, does a latch send bubbles to its neighbors. There are four types of stalling states: N, R, L and B, representing four different bubble sending situations, as shown in Table 1. Notice that the combination of states of all individual latches is the state of the entire circuit.

W	Working with no timing violation
E	Working but a timing violation happens
N	Stalling without sending bubbles
R	Stalling while sending a bubble to its RN (Right Neighbor)
L	Stalling while sending a bubble to its LN (Left Neighbor)
B	Stalling while sending bubbles to both its RN and LN

Table 1. Possible states of a bubble razor latch.

Given the definition of circuit state, the transition rule for a latch's state is the basis of the transition rule of the entire circuit. Table 2 shows the transition rule for a latch in a Bubble Razor ring. The next state refers to the state in the next clock cycle and is related to its neighbors' states instead of its own current state because whether a state stalls and to whom it sends bubbles only depends on timing violation in its Left Neighbor (LN) and the bubbles sent to itself.

If the LN is in state E (rows 1 and 2 in the table), then the stage stalls and propagates bubbles to all of its neighbors who do not send bubbles to it. Rows 3 to 5 represent cases when it receives bubbles from neighbors while there is no timing violation in the LN. If the LN is in state N or L and the RN is in state W, E, N or R (Row 6), then no bubble is sent to this latch. Moreover, its LN is stalling and the instruction arriving at this latch has already been given additional time to ensure its correctness, so its state in next clock cycle must be W. That is, it is impossible for the latch to have a next state of E. On the contrary, if LN is in state W and RN is in state W, E, N or R (Row 7 and 8), it only takes half a clock cycle for the instruction to pass from LN to the current latch and thus both W and E states are possible. The variable p is used to represent the possibility of being in E in this case. In fact this is the only source of uncertainty in our Markov Chain Model.

Given this the transition rule for a latch, the transition rule for the whole circuit is easily derived. Associating each circuit state with a specific integer, the circuit state transition rule could be expressed as a Transition Probability Matrix T of which both the number of row and column equal to the number of possible circuit states. It is worth mentioning that since even and odd latches operate in different clock phases, i.e. half latches don't change in each clock phase, *T* is really a product of two transition matrices each of which represents the transition in each clock phase.

LN's state	RN's state	Next State
E	W, E, N, R	B
	L, B	L
B, R	W, E, N, R	R
	L, B	N
W, N, L	L, B	L
N, L	W, E, N, R	W
W	W, E, N, R	W with probability of *(1-p)*
		E with probability of *p*

Table 2. State transition rule for a bubble razor latch.

After deriving the transition probability matrix for the Markov Chain model, the stationary distribution can be calculated using the equations:

$$\pi = \pi * T$$

$$\sum \pi_i = 1$$

To reduce computational complexity, *T* should not contain transitions to/from impossible states, which are forbidden by state transition rules, or unreachable states, which cannot be reached from an initial all-W state. For N = 1, the optimized size of T is 5 * 5, while for N = 4, the optimized size of T is 449 *449.

B. Performance Analysis

We model the timing cost for error correction with the notion of an *Effective Clock Cycle Time* that is defined as the average time to process each instruction. In particular, according to the Markov Chain model described above, in every clock cycle every latch is either working or stalling and only when it's working can it process an instruction. Consider M clock cycles with a real clock cycle time C with a total time period of $M * C$. The circuit actually cannot process M instructions. Rather, it processes $M * \pi(\text{working})$ instructions, where

$$\pi(working) = \pi(W) + \pi(E)$$

Thus, resulting the effective clock cycle time (EC) can be expressed as follows:

$$EC = \frac{M * C}{M * \pi(working)} = \frac{C}{\pi(working)}$$

It may be insightful to consider the lower and upper bounds on EC. If every combinational cloud delay is shorter than half a clock cycle time (0.5C), no timing violation happens. This means π(working) = 1 and consequently the lower bound on EC = C. The upper bound on EC occurs when all combinational cloud delays are longer than 0.5C, but shorter than C to guarantee the circuit's correctness. In this case, π(working) = 0.5 because every latch of the circuit stalls and works alternately, making EC = 2C.

Now consider the case between the upper and lower bounds. A simple approach to analyzing this case is to ignore the annihilation of bubbles caused by different timing violations. Thus every timing violation causes every latch to stall exactly once. Assume the probability of timing violation for each latch is *p*, the probability of timing violation for one instruction in an N stage circuit, requiring an additional clock cycle to process this instruction, could be estimated as $1 - (1-p)^{2N}$. Thus the EC can be estimated as:

$$EC = C[2 - (1-p)^{2N}]$$

This simplified model overestimates the EC because annihilation of bubbles caused by different errors can occur, reducing the probability of stalling especially for a high probability of error *p*. Our experimental results quantify how much this model overestimates the EC.

C. Delay Distribution

Based on the Markov Chain model and performance analysis above, EC can be expressed as a function of C (real clock cycle time), *p* (probability of timing violation for a latch) and N (number of stages referring to 2N latches in BR or N registers in traditional register-based circuits). It's obvious that *p* is influenced by C. The variable d is used to represent the real delay of a step, i.e., the logic delay from one latch to its Right Neighbor. So *p* can be expressed as follows:

$$p = Probability(d > \frac{C}{2})$$

When considering process variation and aging, the variable d is a random variable with some distribution. In this paper, we consider two different distributions – the first is normal and the second log-normal. Both require only two variables to describe it, i.e. mean μ and standard deviation σ, but the log-normal distribution has a heavy tail that has a basis in the underlying technology in the near-threshold domain [9][11].

In particular, we will explore the benefits of BR with different amounts of variability, as quantified by different σ/μ ratios. The larger this ratio, the larger the relative variability. However, when comparing BR circuits to traditional synchronous circuits, i.e. circuits in which there is no dynamic error correction mechanism, we must also compare distributions for circuits that have different delay lengths, which are correlated to different mean delay lengths μ. Fortunately, reference [12] observes that for die-to-die variations σ/μ ratio is almost a constant for different logic depths, i.e. different delay lengths. For circuits with significant within-die variation, on the other hand, σ/μ ratio decreases for paths of increasing gate lengths, i.e., μ (e.g., see [12]). Moreover to analyze the lower bound of C, it is also important to find out the distribution of the sum of two normal/log-normal variables. References [10] [11] prove that it is reasonable to use another normal/log-normal variable to represent the sum of two normal/log-normal variables.

D. Systematic Error Rate

It is important to recognize that C cannot be too small because we must guarantee that every actual delay between adjacent latches must be shorter than one clock cycle or the additional timing compensation would not be sufficiently long to ensure correctness. Since normal/log-normal distribution does not have an upper limit, we set a rule that the systematic error rate (SER) should be smaller than some small fixed amount. For example, in our results, we assume SER ≤ 0.1%.

When comparing BR circuits to their traditional circuits, we ensure that their SER is the same. For traditional circuits, SER is calculated as:

$$SER = 1 - [Probability(D \leq C)]^{N} \leq 0.1\%$$

where D is a random variable with a mean twice as much as that of d, the delay between neighboring latches in BR circuits. For BR circuits, we note that:

$$SER = 1 - [Probability(d_1 + d_2 \leq 2C)]^{2N} \leq 0.1\%$$

where variable d_1 refers to the delay of this's latches step and d_2 is the delay for the subsequent step in the RN. The inequality below is correct if d_1 and d_2 obey the same normal or log-normal distribution independently.

$$Probability(d_1 \leq C) \leq Probability(d_1 + d_2 \leq 2C)$$

Thus the formula below provides a conservative estimation of the lower limit of C.

$$1 - [Probability(d_1 \leq C)]^{2N} \leq 0.1\%$$

IV. Results and Analysis

A. Markov Chain Analysis Results

Using MATLAB and *Mathematica*, the results of the Markov Chain modeling and performance analysis can be derived. In particular, for small N, the EC can be expressed as a closed-form function of C and *p* as shown in Table 3.

N	$\frac{EC}{C}$, Markov Chain Model	$\frac{EC}{C}$, Simplified Model
1	$\frac{1+3p}{1+p}$	$2-(1-p)^{2N}$
2	$\frac{1+8p+3p^2}{1+4p+p^2}$	
3	$\frac{1+15p+21p^2+3p^3}{1+9p+9p^2+p^3}$	
4	$\frac{1+24p+72p^2+40p^3+3p^4}{1+16p+36p^2+16p^3+p^4}$	

Table 3. Effective cycle time of an N-stage bubble-razor ring.

The lower and upper bounds of EC are exactly the same as what we analyzed above. When probability of error *p* varies from 0 to 1, the effective cycle time ratio EC/C changes from 1 to 2 in response. In addition, as shown in Table 3 and Figure 2, for low p, the effective cycle time is approximately equivalent to 2pC. This shows that the simplified model is a conservative approximation that is particularly close when the probability of error *p* is small.

B. Performance Analysis Results

Considering both normal and log-normal delay distributions, we can extend the MC results and make EC a function of C, μ, σ, and N. Then,

$$K = EC(BR\ circuit) \ /\ EC(Traditional\ circuit)$$

reflects the performance benefit obtained from replacing a traditional circuit with its BR equivalent with the same systematic error rate restriction.

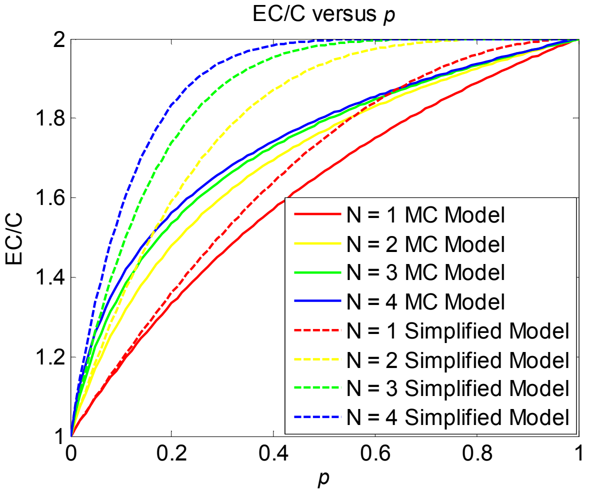

Figure 2. Effective cycle time versus probability of error.

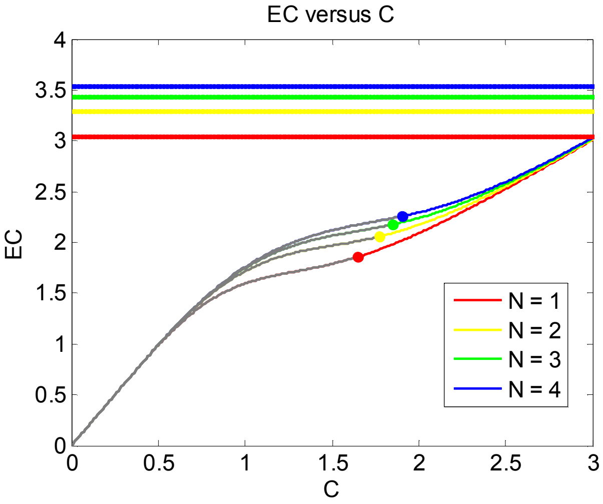

Figure 3. Effective cycle time versus clock cycle time for different ring lengths using a log normal delay distribution.

To simplify our analysis, and with no loss in generality, we set μ = 0.5. Then, based on our Markov Chain results, Figure 3 shows how C influences EC under different numbers of stages when log-normal distribution are used with σ/μ = 0.4. The horizontal lines represent the performance of traditional synchronous design restricted by the same systematic error rate. The EC versus C curves have a slope of 2 for a small C, reflecting that C is too small that errors always occur so that EC is twice as long as C. The curve's slope is 1 for a very large C when the circuit is error-free. It is obvious that we should choose the point with the least EC while ensuring C is above its lower bound. The portions of the curves below the lower bound are colored grey, indicating they are not feasible. The optimum points are marked as large dots in the Figure 3 and represent the operation points that achieve the best performance of the BR circuit for various ring lengths.

From this figure, we can conclude that BR circuits can indeed provide a better performance than traditional design methodology. But, with increasing number of stages, the benefit becomes increasingly small (K increases) because of too many timing violations. Fortunately the change is not significant. Given $\sigma/\mu = 0.4$, K only increases from 0.586 to 0.598 (about 2%) when N increases from 1 to 4. As shown in Figure 4, the results using a normal delay distribution with $\sigma/\mu = 0.4$ have similar trends.

Figure 5 shows how EC varies with C under different delay variances of a log-normal distribution using our MC Model with N = 4. As before, the horizontal lines refer to the performance of the traditional counterparts with the same SER. It shows that BR circuits generally have a shorter effective clock cycle time. For example, for N=4 and $\sigma/\mu = 0.5$, after converting traditional circuits to BR circuits, C is reduced by 45.5% and EC is reduced by 40.2%. This means that the frequency is raised by 83.6% and the effective frequency is improved by 67.3%, which are all significant improvements. Notice that, as expected, as delay variances increase, the benefits from the Bubble Razor increases.

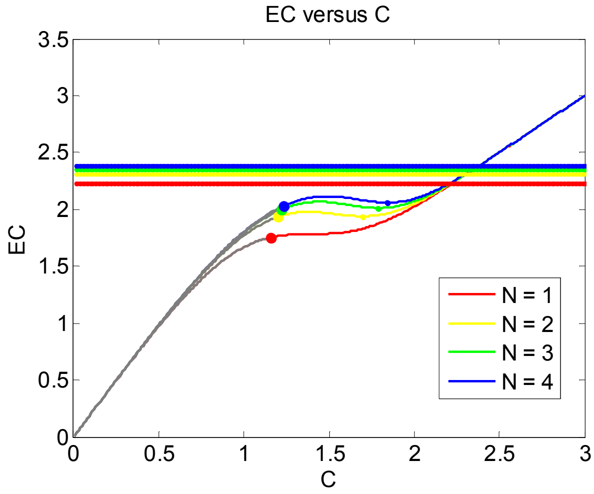

Figure 4. Effective cycle time versus clock cycle time for different ring lengths and a normal delay distribution.

It is meaningful to compare these results with the performance improvements obtained in the ARM Cortex-M3 processor in which Bubble Razor was evaluated. According to [8], the silicon test was based on a pipeline including 170 clusters, which is more far complex than a 4-stage ring. The frequency margin on the traditional synchronous design was set to account for a 10% voltage margin, a 2 sigma process margin, and a 5% additional safety margin. Working at the Point of First Failure (PoFF), the effective frequency improvement over different chips was measured to be between 45% and 82%. When run beyond the PoFF, the effective frequency improvement rises to approximately 100%,which is higher than the 67.3% obtained from our analysis above. One possible cause of this difference is that the effective impact of all margins added is significantly larger than $\sigma/\mu = 0.5$.

Figure 6 shows the cases where N = 4 and $\sigma/\mu = 0.25$ for different models and distributions. According to the figure, different distributions provide quite different results. In particular, the results highlights the fact that log-normal distributions have heavy tails and thus requires a far longer traditional clock cycle time to achieve the same SER as BR circuits.

But different models provide similar shapes. Our experiments show that for log-normal distributions, the difference of the optimum EC derived from the simplified model and MC model does not exceed 5%. The optimum point of our log-normal distribution analysis corresponds to a probability p that is smaller than 25%. This is to be expected because a high p leads to many timing violations that worsen circuit performance. Moreover, when p is relatively low, the simplified model and the Markov Chain model provide similar results. This is important because the simplified can be extended to more complex circuit structures quite naturally. Moreover, it is guaranteed to provide a practical conservative estimate of the circuit's performance.

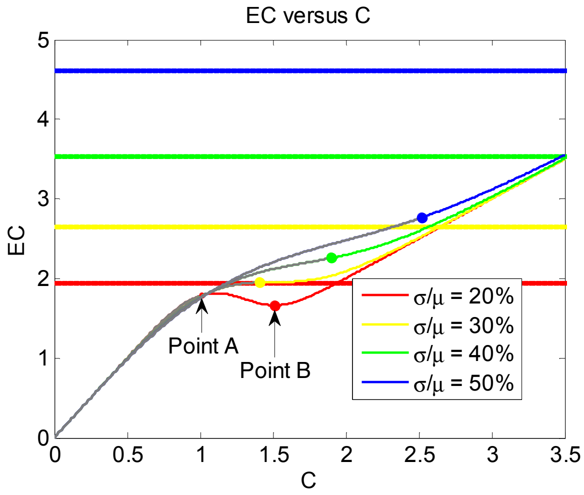

Figure 5. Effective cycle time versus clock cycle times for different variances.

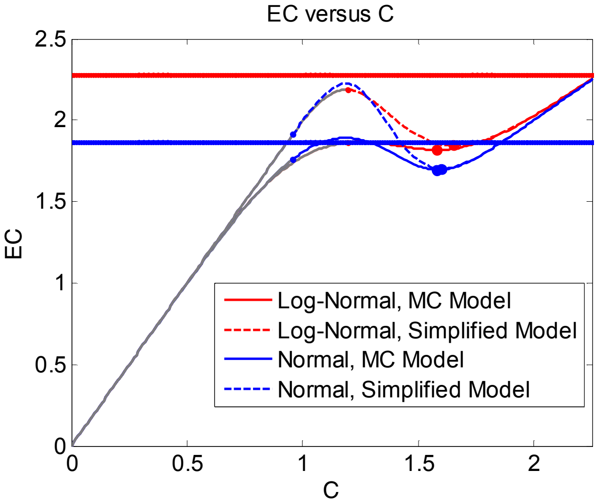

Figure 6. Impact of different models and distributions.

C. *Optimizing strategy for BR circuits*

To simplify our exposition when considering the optimization of BR circuits, we focus on a MC model and log-normal delay distributions. However, our analysis approach applies equally well for other models.

In particular, as shown in Figure 3 and Figure 5, there are two possible optimum points to minimize EC while ensuring the correctness of the circuit. One point, referred to as point A, is determined by the lower bound of C set by the SER limit of the BR circuit. In particular, experiments show that for the ring with N < 5, if $\sigma/\mu \geq 16\%$, setting real clock cycle time to its lower limit, where the systematic error rate is guaranteed not to exceed 0.1%, leads to a lower effective clock cycle time than the traditional design. The other point, referred to point B, is the local minimum point of the EC versus C curve. In particular, for N < 5, experiments show that if $\sigma/\mu \geq 3\%$, setting the real clock cycle time to point B leads to better performance than the traditional design. Note that for very large delay variances, point B does not exist.

It is interesting to compare these two points. Additional experiments when N < 5 show that when the delay variance is small ($\sigma/\mu \leq 28\%$), point B leads to better performance while choosing point A is wiser if the delay variance is large ($\sigma/\mu \geq 31\%$).

Consequently, a way to force the circuit to operate at point A is to iteratively reduce the real clock cycle time until the measured systematic error rate rises just above your SER threshold, e.g., 0.1%. In practice, this point may be the maximum frequency at which all timing errors are successfully corrected. This method gives the ring better performance than traditional design when $\sigma/\mu \geq 16\%$. Moreover, when $\sigma/\mu \geq 31\%$, this method is optimal as it is also better than point B.

If we only use the above method, however, when $\sigma/\mu < 16\%$, BR circuits may seem worse than traditional designs. For BR rings with N < 5, however, if we instead set the real clock cycle time at point B, we can achieve better performance than traditional designs as long as $\sigma/\mu \geq 3\%$. Fortunately, the delay variation often exceeds 16% and thus reducing clock cycle time until the SER limit is hit is, in fact, an effective way to optimize the benefits of BR circuits.

V. Conclusions

This paper uses a novel Markov Chain Model to analyze the performance of bubble-razor circuits. We compare their performance with traditional circuits under both normal and log-normal delay distributions assuming the same systematic failure rate requirement. The results provide a theoretical basis that shows BR circuits have better performance than their traditional counterparts, especially when delay variance is high. The results also show setting clock cycle time as short as possible while still meeting the systematic error rate is usually an effective way to explore the performance potential of BR circuits.

The Markov Chain analysis we provide is accurate but its analysis is complex which limits its applicability to only simple structures. Fortunately, because timing violations happen rarely, the simplified model we propose is not only a conservative approximation but also a close approximation to the Markov Chain model. This is encouraging because this simplified model naturally extends to other more complicated circuit structures.

Finally, note that Bubble-razor circuits have the dual benefit of improving the performance by fixing the voltage and increasing frequency or fixing the cycle time and reducing the voltage supply, thus saving energy. Moreover, more complex applications that trade one benefit for the other are also possible. We have presented a theoretical analysis and an optimization method for optimizing performance. Extensions of this work to also consider energy consumption is an interesting area of future work.

VI. References

[1] M. Nakai, S. Akui, K. Seno, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano and M. Shimura, "Dynamic voltage and frequency management for a low-power embedded microprocessor," *IEEE J. Solid-State Circuits,* vol. 40, no. 1, pp. 28-35, 2005.

[2] K. Hirair, Y. Okuma, H. Fuketa, T. Yasufuku, M. Takamiya, M. Nomura, H. Shinohara and T. Sakurai, "13% power reduction in 16b integer unit in 40nm CMOS by adaptive power supply voltage control with parity-based error prediction and detection (pepd) and fully integrated digital LDO," in *IEEE ISSCC*, 2012.

[3] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner and T. Mudge, "Razor: A low-power pipelined based on circuit-level timing speculation," in *Proc. 36th IEEE/ACM Int. Symp. Microarchitecture (Micro-36)*, 2003.

[4] J. Park, A. Chaudhari and J. A. Abraham, "Non-speculative double-sampling technique to increase energy-efficiency in a high-performance processor," in *DATE*, 2013.

[5] S. Kim, I. Kwon, D. Fick, M. Kim, Y.-P. Chen and D. Sylvester, "Razor-lite: A side-channel error-detection register for timing-margin recovery in 45nm SOI CMOS," in *ISSCC*, 2013.

[6] C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. Bull and D. Blaauw, "Razor II: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 1, pp. 32-48, 2009.

[7] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. Harris, D. Blaauw and D. Sylvester, "Bubble Razor: An architecture-independent approach to timing-error detection and correction," in *ISSCC*, 2012.

[8] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. Harris, D. Blaauw and D. Sylvester, "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction," *IEEE Journal of Solid-State Circuits,* vol. 48, no. 1, pp. 66-81, 2013.

[9] B. Zhai, S. Hanson, D. Blaauw and D. Sylvester, "Analysis and Mitigation of Variability in Subthreshold Design," in *ISLPED*, 2005.

[10] Y. S. Schwartz and S. C. Yeh, "On the distribution function and moments of power sums with log-normal components," *Bell Syst. Tech. J., vol. 61, no. 7,* 1982.

[11] A. Chandrakasan and J. Kwong, "Variation-driven Device Sizing for Minimum Energy Sub-threshold Circuits," in *ISLPED*, 2006.

[12] T.-T. Liu and J. Rabaey, "Statistical Analysis and Optimization of Asynchronous Digital Circuits," in *18th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2012.