School of Electrical, Electronic & Computer Engineering

Modified Bisection Search for Faster Metastability Characterization

Ghaith Tarawneh, Alex Yakovlev

Technical Report Series

NCL-EECE-MSD-TR-2010-154

June 2010

Contact:
ghaith.tarawneh@ncl.ac.uk

alex.yakovlev@ncl.ac.uk

Supported by EPSRC grant EP/G066361/1

NCL-EECE-MSD-TR-2010-154

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Modified Bisection Search for Faster Metastability Characterization

Ghaith Tarawneh, *Member, IEEE,* Alex Yakovlev, *Senior Member, IEEE*

***Abstract*—Circuit state bisection is a robust technique to characterize the performance of multi-stage synchronizers. The passage of metastability between synchronizer stages introduces effects that are not captured by small-signal models and thus numerical integration remains the most reliable method for characterizing this behavior. However, the large number of transient simulations required to characterize one circuit through bisection makes it very difficult to use this technique to run variability analysis or parametric optimization on synchronizer circuits. We present a modified bisection search algorithm that performs 2.5 to 3.2 times faster than conventional bisection without any loss of accuracy. Our method is not restricted to any latch topology or behavior and can safely replace conventional bisection for characterizing any synchronizer circuit.**

***Index Terms*—synchronizers, metastability, bisection.**

I. Introduction

Synchronizers play an important role in regulating the passage of data between systems operating in different clock domains. The increasing scale of device integration and the usage of IP blocks in designs are creating multiple clock domains even within single chips. Synchronizers are thus starting to have a larger impact on within-system communication links and overall system performance.

The arrival of data from a different clock domain often brings the cross-coupled gates in a synchronizer latch to a logically-invalid metastable state that can take a long time to resolve[1]. If such states were not resolved before the arrival of the following clock edge, metastability can cross to other circuits resulting in unpredictable behavior and possible system failure. The probability of this is greatly diminished by cascading a number of latches in a synchronizer to allow any occurrence of metastability to resolve before reaching other logic circuits. However, this introduces latency and adversely affects the performance of the communication link. Since it is impossible to completely prevent metastability, it is important to decide how much time to allow for the synchronizer to settle to meet performance requirements while reducing the probability of synchronization failure to an acceptable level.

Every synchronizer has an absolute metastability point the closer the input to which causes the synchronizer to take longer and longer times to resolve. To characterize the performance of a synchronizer circuit, it is necessary to evaluate how much does its settling time (t_s) increase as the difference (denoted the input window size ΔT_{in}) between the input event time and the absolute metastability point gets smaller. Typically, the design must satisfy a predefined Mean Time Between Failures (MTBF) which corresponds to a certain ΔT_{in}value such that:

$$MTBF = \frac{1}{\Delta T_{in} \times f_c \times f_d} \tag{1}$$

Where f_c is the clocking frequency for the synchronizer and f_d is the data transfer rate.

Previous synchronizer characterization methods involved on-chip measurement techniques such as the ones used in [2], [3] and [4], where the rate of metastable events was "amplified" using control circuitry. Such methods have allowed very rare deep metastable events to be reproduced experimentally but still do not provide sufficiently large MTBF estimates. Designers have thus been relying on small signal models of latch circuits to guide their designs. Using small-signal modeling, it has been established that t_s increases exponentially as ΔT_{in} gets smaller. More formally, their relationship is governed by:

$$\Delta T_{in} = T_w . e^{\frac{-t_s}{\tau}} \tag{2}$$

Where ΔT_{in} and τ are circuit parameters.

The behavior of single-latch synchronizers adheres closely to this model and different synchronizer circuits are often compared according to their τ values which can be obtained by simulation[3]. However, this model does not account for changes in input signals or for large swings that occur as metastability transitions from one stage to another in multi-stage synchronizers. These factors can cause unexpected behavior in multi-stage synchronizers and their effects are not yet fully understood[5]. Measurements on multi-stage synchronizer circuits have shown that the actual MTBF values can be significantly different that those predicted by small-signal models[6][4].

Numerical integration remains the most reliable method to assess the behavior of synchronizer circuits. Typically, two initial input transition times are located such that the synchronizer settles high for one and low for the other. Bisection search is then used to minimize this input timing interval around the synchronizer's absolute metastability point, obtaining progressively smaller window sizes and their corresponding settling times. This method is very reliable but can't be continued for long as the numeric resolution of the input time and node voltages' representation is soon breached and the two states become indistinguishable. In double-precision floating-point representation, this occurs around $\Delta T_{in} = 10^{-20}$ seconds. For f_c=f_d=1Ghz, this is enough to characterize the synchronizer up to a MTBF of only 100 seconds which is well short of common requirements.

II. Overview of Circuit State Bisection

Reference [7] has provided a method to overcome the limitations of both small signal analysis and numerical integration in simulating synchronizers in deep metastable states. Their method is based on the observation that a circuit's dynamic

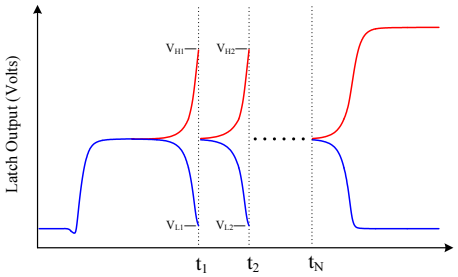

Fig. 1. Circuit State Bisection

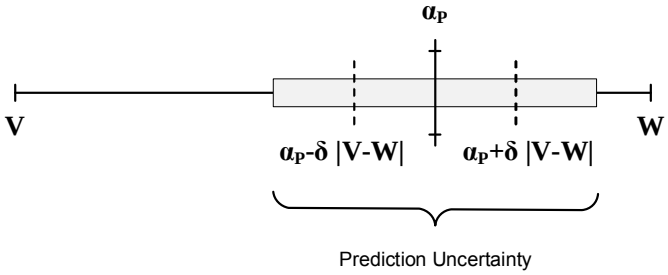

Fig. 2. Bisection Interval

state can be accurately estimated using a linear model when the circuit is operating sufficiently close to a metastable state. A small input time window [t_L,t_H] is first located such that the synchronizer falls in metastability but resolves low when simulated with t_L input delay and high when simulated with t_H. Subsequently a time t_1 is chosen such that the circuit isn't far from metastability in both t_L and t_H simulations, but far enough so that the two sets of simulation traces are clearly distinguishable by the numeric resolution of the integrator. It is then assumed that the difference between the voltage states obtained by the two simulations at t_1 ($|V_L - V_H|$) corresponds to the input window size ($|t_L - t_H|$).

Following that, circuit voltage states between V_L and V_H are interpolated, simulated and their settling times recorded. The associated window sizes for these setting times are obtained by mapping voltage states back to the original window [V_L,V_H] whose time length is assumed equal to $|t_L - t_H|$ in seconds. The interpolation is done through bisection; at every iteration, the middle point of [V_L,V_H] is simulated and becomes either the new V_L or the new V_H. When the difference between the states V_L and V_H becomes too small, they are allowed to diverge and are then captured at a later time t_2. Bisection then proceeds on these more clearly separated states. This process is carried on until a ΔT_{in} corresponding to a sufficiently large MTBF is reached. Figure 1 shows an illustration of this algorithm. This method overcomes the problem of representing the extremely tiny differences between the interval ends of the input window. This is achieved by mapping this difference to a difference in the circuit's voltage states which can be continuously amplified. Using this approach, it is possible to characterize a synchronizer down to extremely small ΔT_{in} corresponding to MTBF values of hundreds of years.

It is possible to implement this method in current simulation tools to compare different synchronizer designs. It can also be used to perform parametric analysis to establish the effects of transistor sizes and various parameters on the behavior of synchronizer circuits. Unfortunately, the production of a single ΔT_{in} vs. t_s plot requires a large number of transient simulations. The reduction of an initial input window of 1ns to 10^{-50} seconds requires $log_2(10^{41})$ or about 137 transient simulations. This is a considerable processing effort even for modern computers. Since varying one circuit parameter can shift the absolute metastability point and change the behavior of the circuit, it is necessary to repeat the bisection procedure for every varied copy of the circuit which sometimes cannot be afforded. Furthermore, designers need to know how much time to allow for synchronizers to settle in order to meet a certain MTBF requirement. They are therefore most interested in the bottom end of the ΔT_{in} vs. t_s plot. However, it is impossible to generate one portion of the plot without having to perform all preceding simulations.

These difficulties make it computationally expensive to use this technique to perform parametric optimization and/or variability analysis. We present a novel method to speed up this technique by a factor of 2.5 to 3.2 without any loss of accuracy, reducing its processing overhead and allowing it to be used to carry intensive circuit analysis.

Our technique makes use of a modified bisection search algorithm taking advantage of our knowledge of the exponential relationship between ΔT_{in} and t_s to increase the convergence speed of bisection.

III. Illustration of Proposed Technique

A. Modified Bisection

In half-interval bisection, the search interval is downsized to one quarter of its size every two successive rounds. The interval mid-point is the optimum choice for bisection in the absence of any additional knowledge about the position of the convergence point (α). Bisecting on points other than the interval mid-point is risky as it can achieve larger or smaller convergence depending on whether the further or the closer interval end is relocated to the bisected point. For example, if the bisected point is chosen very close to the Low end of the interval and simulation proved this point to resolve High, it will become the new High end and the interval size is reduced considerably. However, if this point resolves Low, the Low end of the interval will be relocated by a minor amount which will have a small effect on the interval size. Generally, as the bisected point lies further away from the centre of the interval, greater reduction in interval size is possible but at an increasingly lower probability and much less reduction becomes much more probable.

If an estimate about the position of α can be made, however, this can be used to speed up the convergence speed of bisection. To illustrate this, consider Figure 2 which represents a single bisection round to find the convergence point α in an input time interval [V,W]. Suppose at each round we obtain a rough estimate of α which we call (α_P) whose absolute mean error relative to the interval size is (δ) and is small compared to the interval size. We can assume that each end of the search interval moves independently towards α. We label the points

generated by the i^{th} relocation of each of the ends V and W: V_i and W_i respectively. The i^{th} nested interval $[V_i, W_i]$ may form over two or more bisection rounds and is smaller than its predecessor. We can calculate the i^{th} reduction factor per interval (λ_i) as:

$$\lambda_i = \frac{|V_i - W_i|}{|V_{i-1} - W_{i-1}|} = \frac{|V_i - \alpha| + |W_i - \alpha|}{|V_{i-1} - W_{i-1}|} \tag{3}$$

We assume V_i is obtained after W_i without any loss of generality. Since nested intervals are progressively smaller than their predecessors, V_i is much closer to α than W_i such that $|V_i - \alpha| << |W_i - \alpha|$ and equation (3) can be simplified to:

$$\lambda_i = \frac{W_i - \alpha}{V_{i-1} - W_{i-1}} \approx \delta \tag{4}$$

Therefore we can assume that the size of each nested interval is on average equal to δ times the size of its parent.

Assuming the PDF for α_P is symmetric, the probabilities of α_P falling either side of α are equal. Therefore, for n bisection rounds we expect to obtain $n/2$ nested intervals. The reduction factor per single bisection round is thus approximately $\sqrt{\delta}$ as opposed to 0.5 for conventional bisection. If a predictor can provide an estimate α_P with sufficiently small δ, bisecting on α_P would downsize the interval at a rate of $0.5/\sqrt{\delta}$ faster than conventional bisection. Since the number of simulations required to shrink ΔT_{in} below a certain threshold depends on the gradient of the logarithm of ΔT_{in}, the speedup in simulation count is $log(\sqrt{\delta})/log(0.5)$.

B. The Predictor

To create a predictor for our metastability bisection search, we exploit our knowledge of the exponential nature of the function under bisection. At each bisection round, we use N samples of t_s obtained by previous bisections to calculate a time t_P in our current interval around which the samples appear to have exponentially lower values with increasing ΔT_{in}. The exponential behavior around any time point t is evaluated by calculating the correlation between t_s and the logarithm of ΔT_{in} measured relative to t for the previous N samples as follows:

$$f_{eval}(t) = Corr(log(\Delta T_{in}(t)), t_s) \tag{5}$$

In theory, if t equals the absolute metastability point, the relationship between $log(\Delta T_{in}(t))$ and t_s will be perfectly linear with a negative gradient yielding a correlation coefficient of -1. Thus the closer $f_{eval}(t)$ to -1 the better our candidate t appears to be the absolute metastability point. Since correlation coefficients fall in the range [-1,1], this means that the best t will also yield the minimum correlation coefficient. Therefore, we choose $tp = t$ such that $f_{eval}(t)$ is minimized across our interval.

In our implementation, we have utilized Pearson's product-moment correlation coefficient which for any two variables can be calculated as:

$$Corr(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sigma_x \sigma_y} \tag{6}$$

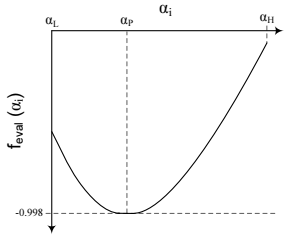

Fig. 3. Illustrative Plot of feval(alpha)

Instead of calculating ΔT_{in} values directly (which requires back mapping of voltage states to the original time window), we express time values relative to the $(N-2)^{th}$ preceding interval and call these α values. The $(N-2)^{th}$ peceeding interval is the smallest that contains the previous N samples at any bisection round. This is because the current interval always contains two t_s samples (the interval ends) and each preceeding interval contains all samples of enclosed intervals plus an additional sample. Since correlation is insensitive to scaling, we can safely substitute ΔT_{in} for $\Delta\alpha$ in equation (5). Figure 3 shows an illustrative plot of $f_{eval}(\alpha)$ across an interval $[\alpha_L, \alpha_H]$. To obtain α_P (corresponding to t_P), it is necessary to iterate through all possible α values in our current interval and find the one that minimizes f_{eval}. It is also necessary to do so with a very small resolution (less than 1/10k of the current interval length in our implementation) because tiny α differences can result in considerable logarithmic differences and can thus vary our choice of α_P considerably. However, evaluating 10k correlation coefficients per bisection round is computationally intensive and can waste the time savings achieved by this technique. To address this, we make use of the fact that f_{eval} is practically single-peaked and use progressively-increasing resolutions to locate its tip. The algorithm we use is presented below:

α_L = low end of current interval relative to the $(N-2)^{th}$ preceding interval
α_H = high end of current interval relative to the $(N-2)^{th}$ preceding interval
while resolution target not met **do**
  STEP $= (\alpha_H - \alpha_L)/M$
  α_i = Generate points in [α_L, α_H] with STEP increments
  α_P = α_j such that $f_{eval}(\alpha_j) \leq f_{eval}(\alpha_i)$ for every i
  α_L = α_P−STEP
  α_H = α_P+STEP
end while

For M=25, it is sufficient to iterate through this loop four times to obtain the best α_P down to a resolution of less than 1/10k of the current interval using only 100 iterations. Smaller values of M will increase the savings achieved by this method but at the expense of increased probability of missing the function's peak. In our simulations, M=25 proved to be sufficiently small and yet safe to use.

IV. Implementation Challenges

Our technique relies on the exponential rising nature of t_s as ΔT_{in} gets smaller. There are circumstances, however, where

this exponential nature can be interrupted for a number of simulations or worse become totally absent. Predictions can then be inaccurate at best, or become completely unreliable. In these cases, bisection will proceed at a very low pace or can be stalled for a number of iterations without progress. Therefore, it is important to detect these states and take suitable measures. In this section, we discuss the two prominent states where the ΔT_{in} vs. t_s plot does not abide by the small-signal exponential model and present solutions to overcome them.

A. Concealed Metastability

In a multi-stage synchronizer, metastability is first observed when the pre-final latch becomes opaque and the final latch becomes transparent. Up until this happens, metastability will propagate through the first stages of the synchronizer without affecting the settling time of the final latch. On a ΔT_{in} vs. t_s plot, this time period appears as a vertical line extending down to a certain ΔT_{in} value where metastability first appears at the synchronizer output. Since the settling times of the final latch do not change during this period, it is impossible to use them to predict α_P values for bisection. In a multi-stage synchronizer, this segment represents a major part of the plot and without being able to utilize the technique in this part the overall speedup is greatly reduced.

To overcome this, we track metastability as it propagates through the stages of the synchronizer and use intermediate settling times of stages to make our predictions. While metastability in stage 1 of the synchronizer is not visible on the output of stages 3 and after, it still affects the settling times of stages 1 and 2. Instead of recording the settling time of the final stage, we do so for all stages in every bisection round. We maintain a table of settling times and use it along with the input clock timings to determine which stages have become metastable. Samples from metastable stages are then used to calculate our α_P values.

In our simulations, we have found that a near-perfect metastable state in one latch does not induce metastability in the next. Instead, metastability is propagated when the previous latch begins to resolve just before the clock edge. The dynamics of the preceding latch (which may have diverged sufficiently from metastability) just at the arrival of the clock edge control the passage of metastability to the next stage together with how long it lasts. Therefore, continuous bisection on the state of the final latch of the synchronizer converges on the input time instance where intermediate stages resolve in a perfect timing to induce the longest possible metastable state in the last stage. This input time instance is different from these which cause the intermediate stages to remain in metastability for longer times and so every latch in the synchronizer has its own absolute metastability point.

The decision of which interval end to relocate to the bisected point must depend on resolving state of the final stage. This is necessary to ensure that the algorithm will converge on the absolute metastability point of the final stage and not on those of previous stages. Using the t_s samples obtained from a stage K, predictions will point towards the metastability point (which is a resolution-limited representation of the absolute metastability point) of stage K. However, this point will be the same as the metastability point of the final stage up until metastability propagates to stage K+1. In other terms, the absolute metastability point of stage K is indistinct from that of the last stage down to the resolution of ΔT_{in} at which metastability propagates to stage K+1. As bisection on the state of the final stage continues, the High and Low interval ends start to make stage K resolve to the same state (either High or Low) and the settling times observed at this stage reach a maximum ($t_{max}(K)$). Since circuit state bisection is regularly advanced to new times in the transient simulation, it becomes impossible to observe the settling time for stage K when the transient simulation is started from time instances exceeding $t_{max}(K)$. On a ΔT_{in} vs. multi t_s plot (where ΔT_{in} is measured relative to the absolute metastability point of the final stage), the settling times for intermediate stages curve downwards approaching their maximum values before disappearing as metastability transitions to the next stages. This effect is portrayed in Figure 4.

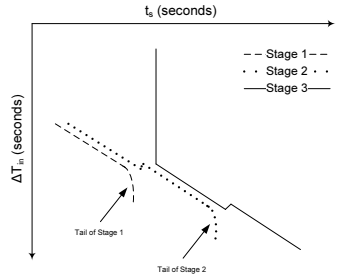

Fig. 4. Curving Down of Settling Times for Intermediate Latch Stages

B. Clock Back Edge Effect

The crossing of metastability from one stage of the synchronizer to another tends to introduce an offset in settling time. This was first observed by Kinniment [2] and is called the clock back edge effect. The offset creates a short disruption in the exponential rising nature of settling times and therefore momentarily decreases the accuracy of the predictor.

Furthermore, the behavior of synchronizers during the occurrence of such disruptions cannot be predicted accurately. Depending on the circuit topology, it may not be possible to observe metastability when the latch state does not change and before the latch falls in deep metastability. This usually happens whenever metastability crosses to a new stage and is joined by the settling times for the previous stage reaching their maximum. In such scenarios, the settling times for the non-changing state of both the current and next latches cannot be recorded, and no enough samples exist to form our prediction. We temporarily disable the prediction routine as soon as we detect this and resort to half-interval bisections for few rounds. This bypasses the hazardous region and then prediction can be resumed.

V. Simulation Results

We have created an automatic tool for the purpose of investigating this technique. While the authors in the original

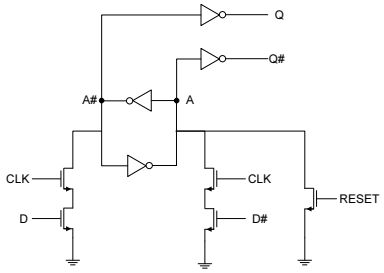

Fig. 5. Schematics for a Typical Latch

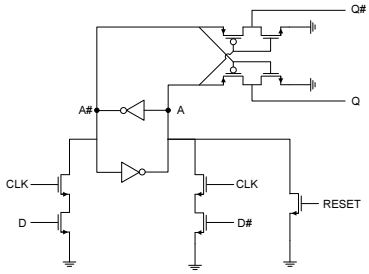

Fig. 6. Schematics for a Filtered Latch

TABLE I
SIMULATION RESULTS FOR TYPICAL LATCH SYNCHRONIZERS

Number of Stages	Transient Simulation Count		Speed Improvement
	Conventional Bisection	Modified Bisection	
1	242	76	3.18x
2	238	86	2.77x
3	241	86	2.80x
4	242	93	2.60x

TABLE II
SIMULATION RESULTS FOR TYPICAL FILTERED SYNCHRONIZERS

Number of Stages	Transient Simulation Count		Speed Improvement
	Conventional Bisection	Modified Bisection	
1	254	79	3.22x
2	247	89	2.78x
3	243	99	2.45x
4	241	97	2.48x

implementation used simple transistor models in MATLAB to perform numerical integration, our tool creates SPICE simulation jobs and uses NGSPICE to simulate synchronizer circuits with more realistic 45nm BSIM4 predictive technology models[8]. Our tool is fully automated and is not restricted to any circuit topology or model technology.

We have applied our technique to circuits consisting of different numbers of latch stages. In our analysis, we used two types of latches; a typical latch and one that is fitted with a metastability filter. We cascade a number of these latches to form our synchronizers and simulate them by applying separate clock pulses to every stage, each 1ns wide with 50ps falling and rising times.

For each bisection run, our tool first uses typical input time bisection to locate an initial input time window whose length is 1ps. Bisection is then transformed into the circuit state space and continued till the settling time for the synchronizer is extended to 6ns. Our tool records the settling times of all latch stages during the simulation and uses these of metastable latches to calculate predictions for bisection. In the absence of enough samples to form a prediction (which happens around clock edges), the tool automatically switches to half-interval bisection till a sufficient number of samples is recollected.

At the end of bisection, the number of transient simulations performed during the analysis is noted. We have repeated this procedure for synchronizer circuits with different number of stages using both conventional bisection and our modified bisection algorithm.

A. Typical Latch Synchronizers

Figure 5 shows the circuit diagram for a typical latch stage. At the start of the simulation, the cross coupled inverters are pulled into a known state by the RESET signal. The latch is then brought into a metastable state by adjusting the arrival of a rising D signal relative to the falling edge of CLK.

The results of performing bisection on synchronizers composed of this latch stage are summarized in Table I.

B. Filtered Latch Synchronizers

We repeated the same analysis for synchronizers composed of latch stages which were fitted with metastability filters[9]. Figure 6 shows the circuit diagram for one such latch. The results for performing bisection on synchronizers made of this latch stage are summarized in Table II.

The average speed up for this circuit is slightly less than that of typical-latch circuits, with the difference becoming more noticeable for the 3 and 4-stage synchronizers. The simulation logs generated by our tool indicated that the settling times of the transparent metastable latches could not be observed when their states did not change. This forced the tool to disable the prediction routine around every clock edge, thus reducing the speedup of the algorithm. The invisibility of metastability when the latches' states were not changed is attributed to the use of the metastability filters.

C. Predictor Performance

To evaluate the performance of our predictor, we calculate the position of the final interval relative to all proceeding intervals and call these the actual α values. Since the final interval is much closer to the absolute metastability point than all previous intervals, it would be safe to assume that the actual α values represent the ideal predictions. Therefore we use these as a reference to calculate the predictor's error. Figure 7 shows a plot of α_P versus actual α for bisection on a single stage synchronizer. The predictions are noticeably close to the actual α values demonstrating a good predictor

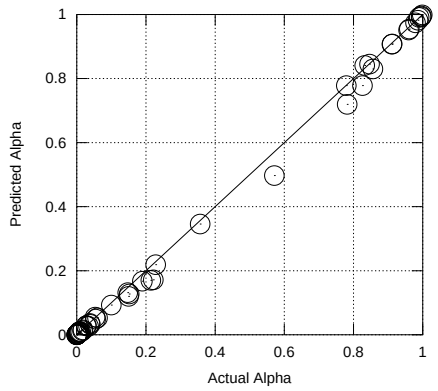

Fig. 7. Predictor Performance

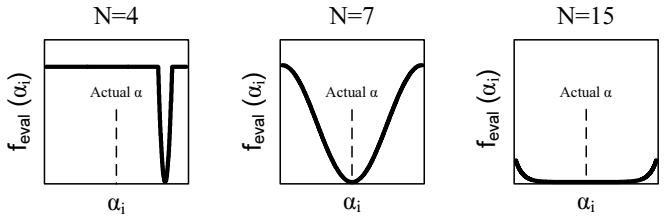

Fig. 8. Effect of choice of N on feval

performance. The α values are also more condensed near the interval ends which indicates that modified bisection has been relocating the interval ends very close to the metastability point at every round. This shows that the algorithm is performing much better than conventional bisection. The absolute mean error of prediction for our single-stage synchronizer is 0.009 relative to the interval size. Using the expression we derived in Section 3, the transient simulation count speedup can be approximated at $log(\sqrt{0.009})/log(0.5) \approx 3.4$x which matches the speedup obtained in our simulation for a single-stage synchronizer (3.18x).

In multi-stage synchronizers, the actual speedup is determined by the ratio of bisection rounds where the prediction routine cannot be used to the total number of rounds. Therefore, for circuits with longer clock pulses and lower clock back edge delays, the algorithm speedup will be more closer to the theoretical speedup of $log(\sqrt{\delta})/log(0.5)$.

To maximize the benefit of our technique, it is important to optimize the performance of the predictor. The most dominant factor in this regard is the number of samples N to include in computing f_{eval}. Larger N values will produce better averages and reduce the effects of spurious samples. However, as more distant samples are included, our choice of α_P in the current interval will appear to have less effect on the correlation coefficients. This is because intervals shrink exponentially, so for a nested interval i, all points within the nested interval $i+10$ for example will appear to be equally suitable to be the exponential curve's center points. If an excessively large N is used, f_{eval} will have a wide peak whose tip is hard to locate. The choice of N is thus a tradeoff between better averaging and a sharper indication of α_P by f_{eval}. These effects are portrayed in the plots in Figure 8.

It is hard to determine a generally optimal N value as this depends on the details of each implementation. In our own, we have found an N value of 7 to produce the most accurate predictor.

VI. Conclusions

We have presented a novel algorithm that is able to characterize the performance of synchronizer circuits with 2.5 to 3.2 times less transient simulation effort. Our method exploits the exponential nature of the relationship between settling times and input window sizes to speed up the convergence speed of bisection. We have shown that bisecting on relatively accurate predictions of the location of the convergence point increase the speed of convergence and that the amount of speedup is directly related to the accuracy of the predictor.

We have used our algorithm to analyze a number of synchronizers composed of two latch stages that exhibit slightly different behavior. In the course of developing our metastability tracking algorithm, we analyzed the behavior of latches as metastability crosses from one to another. Our simulations have shown that bisection on the resolving state of the final stage of synchronizers tends to bring every pre-final latch to resolve just before the clock edge on which the following latch becomes opaque. Therefore, a distinct absolute metastability point exists for every latch in a synchronizer. However, each of these metastability points is indistinguishable from that of the last latch down to the size of input window that cause metastability to cross to the following latch. We exploit this relationship to direct our bisections to the absolute metastability point of the final latch in the absence of changes in its settling time.

Our method is not restricted to any latch topology or behavior and can be improved by enhancing the predictors accuracy. We aim to utilize this method to aid us in carrying more comprehensive investigations of multi-stage synchronizers in the future.

References

[1] T. J. Chaney and C. E. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Trans. Comput.*, vol. 22, no. 4, pp. 421–422, 1973.

[2] D. Kinniment, K. Heron, and G. Russell, "Measuring deep metastability," in *ASYNC '06: Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems*. Washington, DC, USA: IEEE Computer Society, 2006, p. 2.

[3] C. Dike and E. T. Burton, "Miller and noise effects in a synchronizing flip-flop," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 849–855, 1999.

[4] Y. Semiat and R. Ginosar, "Timing measurements of synchronization circuits," in *ASYNC '03: Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems*. Washington, DC, USA: IEEE Computer Society, 2003, p. 68.

[5] I. W. Jones, S. Yang, and M. Greenstreet, "Synchronizer behavior and analysis," in *ASYNC '09: Proceedings of the 2009 15th IEEE Symposium on Asynchronous Circuits and Systems (async 2009)*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 117–126.

[6] B. A. Kinniment D. J. and Y. A. V., "Synchronization circuit performance," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 202–209, 2002.

[7] S. Yang and M. Greenstreet, "Computing synchronizer failure probabilities," in *DATE '07: Proceedings of the conference on Design, automation and test in Europe*. San Jose, CA, USA: EDA Consortium, 2007, pp. 1361–1366.

[8] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *ISQED '06: Proceedings of the 7th International Symposium on Quality Electronic Design*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 585–590.

[9] N. Cline, Ronald L. (Albuquerque, "Method and circuit for improving metastable resolving time in low-power multi-state devices," U.S. Patent 5 789 945, August, 1998. [Online]. Available: http://www.freepatentsonline.com/5789945.html