Abstract—Synchronizers are used to mitigate the effects of metastability in multiple clock domain System-on-Chip devices. In order to enable reliable synchronization, the synchronizer MTBF (Mean Time Between Failures) should be much longer than the product lifetime. To achieve such high margins, multistage synchronizers are used. Several simulation methods have been developed to determine their probability of failure and the number of stages to use. While simulation methods have improved in recent years, accurate analytical models for failure calculations are scant. Some previously published models do not reflect estimations of MTBF but only loosen lower bounds that give rise to a high number of synchronizer stages reducing the overall system performance. Others provide improved accuracy but are difficult to operate and simulation for each stage is required. In this paper, we review published analytical models for MTBF calculations of multistage synchronizers. We show that existing models often underestimate MTBF, and in some cases they even overestimate it. A new model that calculates a MTBF lower bound with significantly smaller margins is introduced. These estimates are shown to be consistent with state of the art simulations and measurements. A method for calculating these estimates for a variety of applications is presented based on a limited number of intrinsic synchronizer parameters determined by simulation.

1. INTRODUCTION

The System-on-Chip (SoC) designer who wishes to use a synchronizer from a standard cell library would like to know the MTBF (Mean Time Between Failures) of the system including the synchronizer before signing off on the design. This knowledge is increasingly valuable in nanoscale SoC designs because several factors have emerged that jeopardize the reliability of synchronizers. In particular, the number of synchronizers in a design is growing rapidly, the variability of semiconductor parameters is troubling as is the sensitivity to operational conditions.

Prediction of MTBF in clock domain crossing (CDC) scenarios, Figure 1, depends on a variety of parameters. Some of these parameters are extrinsic; they are related to how a synchronizer is used in the application at hand. For example, the clock frequency \( f_c \), \( 1/T \), rate of data transitions \( f_d \), clock duty cycle \( \alpha \) and the number of stages in the synchronizer \( N \) are all parameters related to the application.

Other essential parameters are related to synchronizer intrinsic characteristics. The most important of these are the settling time-constants \( \tau_i \) of the synchronizer’s bi-stable stages, \( i = 1, 2 \ldots N \). Also important is the aperture width \( T_W \). These parameters must be determined by physical measurement, or circuit simulation. They are strongly dependent on the characteristics of the semiconductor process and the synchronizer operating conditions, such as supply voltage and temperature.

Figure 1. A typical multistage synchronizer

Finding values for all of these parameters and determining their influence on MTBF is a challenging undertaking. Physical measurement of synchronizer characteristics is usually limited to the very first stage [1][2][3], because of the unbounded time required to carry out measurements on later synchronizer stages. Reliable simulation of the entire synchronizer is now possible, however, due to state of the art simulation methods [4], and has been validated against first stage measurements [5]. Thus, the overall MTBF of a multistage synchronizer can be evaluated by simulation for a selected set of extrinsic and intrinsic parameters.

It is desirable to avoid simulating for various extrinsic parameter combinations. To discern the contribution of each parameter, we seek a formula that calculates MTBF for an arbitrary set of extrinsic parameters, and is based on the set of intrinsic parameters determined from simulations. This approach would still require simulations for each synchronizer circuit, for each transistor model and for each set of operating conditions, but the variations in results arising from changes in extrinsic parameters can be dealt with analytically. Another reason for the importance of an accurate analytical expression is that currently available formulas provide pessimistic lower bounds on the MTBF. The result is a relatively high increase in latency due to unneeded synchronizer stages that degrade the overall performance.

Separation of extrinsic and intrinsic parameters has substantial advantages for both the synchronizer circuit designer and the SoC designer. In today’s silicon IP marketplace these roles are likely to be performed by different individuals who may work for different organizations. Because of the trend toward developing synchronizers as specialized standard cells, only the cell designer may have access to the semiconductor process
models necessary to support estimation of the intrinsic parameters of a synchronizer cell. Similarly, extrinsic parameters depend on the application and are decided by the system integrator or SoC designer. This work develops a formula that separates intrinsic and extrinsic parameters and enables MTBF estimation in multistage synchronizers. The formula is an intuitive expression for MTBF that the SoC designer will find easier to use than most published methods. Section 2 provides a survey of previously published MTBF formulas for multistage synchronizers. In Section 3 we develop the novel formula for multistage MTBF and introduce the concepts of $\tau_{\text{eff}}$ and $T_W(N)$, an effective settling time constant and an effective aperture window. Section 4 deals with the implications and discussion of the model and Section 5 shows simulations that confirm the model. In Section 6 we present some conclusions.

## 2 Published MTBF Models

Several MTBF models have been explored since the discovery of the metastability effect. Table 1 shows a summary of publications that have introduced formulas for multistage MTBF calculations.

The column Formula in Work presents the MTBF formula as it appears in each publication, the MTBF Unified Model column uses a standardized nomenclature in order to compare the expressions more easily. In [7] the term $t_s^2$ represents the average position of the metastability window in the slave input. In [8], $t_{\text{sna}}$ represents the setup time of the latches used in the flip-flops (FFs).

### Table 1. Summary of existing multistage synchronizers MTBF models

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Formula in Work</th>
<th>MTBF Unified Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[6]</td>
<td>1987</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{\tau_{\text{f}}}$</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{\tau_{\text{f}}}$</td>
<td>MTBF for a $N+1$ FF synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[7]</td>
<td>1992</td>
<td>$\alpha_{DF}N+1 \left( \frac{e^{-\tau_{\text{f}}/\tau} e^{N_1/\tau}}{\tau} \right)^{N_{\text{f}}-1}$</td>
<td>$\tau^N e^{N_{\text{f}}/\tau}$</td>
<td>MTBF for a $N$ FF synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[8]</td>
<td>1997</td>
<td>$\frac{t_s^2}{1-t_s^2}$</td>
<td>$\frac{2t_s^2}{1-t_s^2}$</td>
<td>MTBF for a 2 FF synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[9]</td>
<td>2003</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{A_{\text{f}}T}$</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{A_{\text{f}}T}$</td>
<td>MTBF for a $N+1$ latches synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[10]</td>
<td>2007</td>
<td>$\frac{N/T}{T_{Wf1}f_{d}}$</td>
<td>$\frac{N}{T_{Wf1}f_{d}}$</td>
<td>No explicit formula shown.</td>
</tr>
<tr>
<td>[11]</td>
<td>2009</td>
<td>$\left[ f_{si} \Delta t_{\text{in,j}}(T_{\text{sna}}) \prod_{i=1}^{N-1} \Delta t_{\text{in,j}}(T_{\text{sna}}) \right]^{-1}$</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{T_{Wf1}f_{d}}$</td>
<td>$^*\text{Original formula in paper was for } N = 4 \text{ latches}. Result can be extended for } N \text{ latches.}</td>
</tr>
<tr>
<td>[12]</td>
<td>2010</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{T_{Wf1}f_{d}}$</td>
<td>$\frac{N(T-\tau_{\text{fad}})}{T_{Wf1}f_{d}}$</td>
<td>MTBF for a $N+1$ FF synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[13]</td>
<td>2011</td>
<td>$\frac{N/T}{T_{Wf1}f_{d}}$</td>
<td>$\frac{N}{T_{Wf1}f_{d}}$</td>
<td>MTBF for a $N+1$ FF synchronizer. Master and slave latches are assumed identical</td>
</tr>
<tr>
<td>[14]</td>
<td>2012</td>
<td>$\left( \prod_{j=1}^{N-1} \left( \frac{N_j}{T_j} \right)^{N_j} \right)^{1/T_{Wf1}f_{d}}$</td>
<td>$\left( \prod_{j=1}^{N-1} \left( \frac{N_j}{T_j} \right)^{N_j} \right)^{1/T_{Wf1}f_{d}}$</td>
<td>MTBF for a $N$ latches synchronizer.</td>
</tr>
</tbody>
</table>
In [11] \( \Delta t_{in,j} (T_{s,j}) \) represents the data-clock separation at the input of stage \( j \) that generates a settling time of \( T_{s,j} \) at its output. In [14] \( T^W_j \) and \( \tau_j \) represent the time window and the settling time-constant of the stage \( j \). In [10] and [13] the \( MTBF \) of \( N + 1 \) flip-flop stages is proportional to waiting \( NT \) cycle times for metastability resolution. On the other hand, [7], [8], and [14] predict \( T^W \) has an exponential relation with \( N \). In all of the surveyed papers except [11] and [14], the flip-flops in the synchronizer were taken to be identical, and no differentiation has been made between the master and slave latches composing the flip-flops. Formulas [7],[11] and [14] provide higher accuracy compared to the others but their usage is non-trivial and several independent simulations are needed to estimate \( \tau_S, \tau, \Delta t_{in,j} (T_{s,j}) \) and \( T^W_j \) for each stage. The accuracy obtained by those formulas is compromised by the ease of calculation in [10] and [13]. The influence of clock duty cycle is not evident in any of the survey formulas.

To illustrate the mismatch those models predict, we compare the \( MTBF \) calculated using formula [13] with simulations performed using method [4]. Figure 2 shows simulated and calculated values of \( MTBF \) for varying number of stages (\( N \)) in the synchronizer. The values of \( \tau \), and \( T^W \) were extracted from simulations of a single flip-flop and the \( \tau \) of the slave (\( \tau = \tau_S \)) was used. All flip-flops in the chain were taken to be identical and their interconnect delay was assumed negligible. The results show a significant mismatch of several orders of magnitude between simulations and calculations. This exemplifies the fact that using simple models generates inaccurate values and, hence, the importance of developing a formula for an accurate estimation of \( MTBF \).

The circuit diagram shown in Figure 3 is the circuit used throughout this paper. The master and slave regenerating inverters are within the dashed lines. The master latch is transparent when the clock (C) is low and captures the data (D) when C goes high. The slave latch is transparent when C is high so the captured D appears at \( Q_M \) a clock-to-Q delay later (\( t_{pu} \)). When C falls, the state of the master is captured by the slave. If, however, D changes during a window of vulnerability near the rising edge of the clock C, \( Q_S \) may fail to be a valid voltage at the next rising edge of C. This presents a metastability hazard and a possible system failure. Failure may occur when \( Q_S \) is not a valid voltage (in the excluded range in Figure 4, e.g., \( V_{Q_S} \in (V_{IL}, V_{IH}) \)). If \( Q_S \) is delivered to multiple flip-flops, some may register a high and others a low logic level. Although all of these flip-flops may each have valid outputs, a system failure may occur because an illegal system state may exist if all versions of \( Q_S \) are not the same.

Figure 4 shows a simulation of a master-slave synchronizer flip-flop exhibiting metastability. In this simulation, D changed close to C causing metastability at \( Q_M \). \( Q_M \) is changing near the falling edge of C causing metastability at \( Q_S \).

### 3 Proposed MTBF Model

We start by analyzing a master-slave flip-flop and then extend the results to a chain of an arbitrary number of flip-flops in the next sub-section.
Figure 5 shows timing diagrams of the outputs of the master and the slave during metastability. The timing diagram shows only the resolution of the outputs, but is useful as an introduction to the theory developed in this section. This theory disregards second order effects such as latch propagation delays, realistic rise and fall times, interstage delays, non-linear effects, setup-time delays and the effects of noise. These realities are addressed in Section 3.5 where it is shown there is no loss in the generality from these simplifications. In the top case in Figure 5, for a data-clock offset (δ) in the red vulnerability window for D, the output Q_M will be resolving at a time near or past the falling edge of C. Specifically, Q_M resolves high for δ > δ_M and low for δ < δ_M as shown by the arrows. In the bottom case, the window of vulnerability is one that causes the output Q_S to resolve near or past the next rising edge of C. As before, Q_S resolves high for δ > δ_S and low for δ < δ_S. This case where Q_S is still metastable at the next rising edge of the clock is when a synchronizing error for the complete flip-flop may occur. The precise data-clock offsets, δ_M and δ_S, are the theoretical values that would produce indefinite metastability in the master and the slave, respectively, and their values are not necessarily the same.

There are two significant observations associated with Figure 4 and Figure 5. One is that while the clock is high, the settling behavior at Q_M is a function of τ_M, the master settling time-constant, and while the clock is low, Q_S is a function of τ_S, the slave settling time-constant. The second observation is that if Q_M is in the vulnerability window for the slave latch as the clock goes low, the metastable behavior at Q_S will ensue.

Three voltage constants and two voltage functions are defined in the analysis of the master-slave chain:

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_M(M)</td>
<td>Metastable voltage at Q_M, generated by time-offset δ_M</td>
</tr>
<tr>
<td>V_M(S)</td>
<td>Metastable voltage at Q_S, generated by time-offset δ_S</td>
</tr>
<tr>
<td>V_e(M)</td>
<td>Metastable voltage at Q_M, causes slave metastability</td>
</tr>
<tr>
<td>V_Q,M(t, δ)</td>
<td>Voltage at Q_M, a function of time t and offset δ</td>
</tr>
<tr>
<td>V_Q,S(t, δ)</td>
<td>Voltage at Q_S, a function of time t and offset δ</td>
</tr>
</tbody>
</table>

Due to noise, perfectly constant metastable voltages V_M(M) and V_M(S), are not physically achievable, but the idea does define the line of separation, or separatrix, between the high-resolving and low-resolving outputs of a latch. As shown in Figure 5, the vulnerability window around δ_M is wider than that around δ_S. The wider window contains the narrower window. Also δ_M is always within the wider window and δ_S is always within the narrower window.

Let the origin of time (t = 0) be at the first rising clock edge, T be the clock period and α be the fraction of T for which the clock is high. After the normal propagation time t_{pd}, before αT and near metastability, the master output at Q_M is linear and for small variations away from V_m(M) the behavior of V_Q,M(t, δ) is given, for t > t_{pd}, by the equation:

\[
V_Q,M(t, \delta) - V_m(M) = G_{tv}\exp\left(\frac{t}{t_{pd}}\right)(\delta - \delta_M) \tag{1}
\]

Here, the variable δ is the data-clock offset in time and δ_M is the particular offset that produces an indefinitely long period of metastability of the master, meaning V_Q,M = V_m(M). Near metastability we assume linearity, that all the circuit variables are continuous and that all the circuit parameters are constant. Therefore, the circuit can be modeled by a set of linear ordinary differential equations. There must be at least one positive root of the associated characteristic equation if there is to be regeneration and the resulting growing exponential behavior. Assume the solution associated with the largest positive root characterizes the eventual circuit behavior and solutions associated with other roots are neglected. The coefficient G_{tv} of this exponential solution is the time-to-voltage gain through the circuit from the input D to the node Q_M and has the units Volt/sec. The value of G_{tv} depends on the origin of time and we define it at the rising clock edge, for convenience. This convention implies that (1) is invalid for \( t < t_{pd} \). The fact that the effect of t_{pd} can be absorbed in the coefficient G_{tv} is suggested in section 3.5.

Near the falling clock edge and for a data-clock offset δ_S at the input to the master, there will be a critical voltage V_Q,S(αT, δ_S) = V_m(S) at the input to the slave that causes marginal triggering of the slave. This vulnerability voltage, V_m(S), becomes significant some time before the falling clock edge at αT, causing the output of the slave, after t_{pd}, to reside at V_m(S) indefinitely. Thus, V_m(S) is the slave separatrix between high and low resolving traces. Assume the setup time is negligible so that an expression similar to (1) for V_Q,S(t, δ) for \( t \in (αT + t_{pd}, T) \) can be written:

\[
V_Q,S(t, \delta) - V_m(S) = G_{tv}\exp\left(\frac{t - αT}{t_S}\right)(V_Q,M(αT, \delta) - V_m(M)) \tag{2}
\]

Later in Section 3.5 we justify how the non-negligible setup time can be covered in this analysis. Linearity of the slave circuit near V_m(S) is used to establish the linearity of (2). The coefficient G_{tv} is a voltage-to-voltage gain between the slave input and the node Q_S. Combining (2) and (1) for \( t \in (αT + t_{pd}, T) \) yields:

\[
V_Q,S(t, \delta) - V_m(S) = G_{tv}\exp\left(\frac{t - αT}{t_S}\right)\left[V_m(S) + G_{tv}\exp\left(\frac{t}{t_{pd}}\right)(\delta - \delta_M) - V_m(M)\right] \tag{3}
\]
After \( t_{pd} \), the data-clock offset \( \delta_2 \) leads to indefinite metastability in the slave and a constant slave output \( V_m(\delta_2) \). This voltage corresponds to the separatrix at the slave for high and low resolving traces. To make (3) independent of time during metastability, the value of \( \delta_c \) must be such that the bracketed expression in (3) vanishes:

\[
V_m(M) + G_{tv} \exp \left( \frac{\alpha T}{\tau_M} \right) (\delta_S - \delta_M) - V_o(M) = 0
\]  

(4)

Subtracting (4) from the bracketed expression in (3) and evaluating at \( t = T \) yields:

\[
V_{QS}(T, \delta) - V_m(S) = G_{tv} G_{vv} \exp \left( \frac{\alpha T}{\tau_M} + \frac{(1 - \alpha) T}{\tau_S} \right) (\delta - \delta_S)
\]  

(5)

From (5), we define \( \delta_* \) as the clock-data separation that yields the voltage \( V_{QS}(T, \delta_*) = V_{ll} \) at time \( t = T \). Likewise, define \( \delta_- \) so that \( V_{QS}(T, \delta_-) = V_{ll} \). It is then possible to calculate the vulnerability window within which a data-clock offset \( \delta \) will produce an invalid output,

\[
\delta_* - \delta_- = \left( \frac{V_o}{V_{QS}(T, \delta_*) - V_{QS}(T, \delta_-)} \right) \exp \left( \frac{\alpha T}{\tau_M} + \frac{(1 - \alpha) T}{\tau_S} \right) \frac{G_{tv} G_{vv}}{G_{tv}}
\]  

(6)

The coefficient \( V_o \) defines the voltage difference between borderline valid voltages at the output of the second latch. Above, below and at these voltages the slave will cause no marginal triggering of any following flip-flops. Note that \( V_o / G_{vv} \) is the voltage window of vulnerability at the input to the slave.

For a uniform distribution of data-clock offsets \( \delta \) over the clock period \( T \), the probability of failure is bounded by:

\[
Pr(fall) \leq \frac{\delta_* - \delta_-}{T}
\]  

(7)

All data-clock offsets inside the metastability window \((\delta_* - \delta_-)\) will generate traces with voltages within an output window whose size is \( V_o \) at \( t = T \) and hence are prone to produce metastability in following stages. Since the details of the next stages may be unknown, not all traces in this window will actually produce metastability in a following stage. Hence, the inequality represents an upper bound on the failure probability. (For now, we assume the availability of a full clock period of settling time. Logic delays, multiple destinations or long wires may interfere with that assumption and such circumstances will be addressed in section 3.3.)

From (7), with a data transition rate \( f_d \), the \( MTBF \) is:

\[
MTBF = \frac{1}{Pr(fail) f_d} \geq \frac{T}{(\delta_* - \delta_-) f_d} = \frac{G_{tv} G_{vv} T}{V_o f_d} \exp \left( \frac{\alpha T}{\tau_M} + \frac{(1 - \alpha) T}{\tau_S} \right)
\]  

(8)

To make (8) resemble the familiar formula for \( MTBF \) of a single latch, we define an effective settling time constant:

\[
\tau_{eff} = \left( \frac{\alpha}{\tau_M} + \frac{(1 - \alpha)}{\tau_S} \right)^{-1}
\]  

(9)

The lower bound on the \( MTBF \) of a master-slave flip-flop (8) then becomes:

\[
MTBF \geq G_{tv} G_{vv} T \exp \left( \frac{T}{\tau_{eff}} \right)
\]  

(10)

3.2 \( N \) Concatenated Flip-flops

In (10) the lower bound on the \( MTBF \) of a single master-slave flip-flop can be calculated. To extend this result to a chain of \( N \) flip-flops, the process described in (1) to (8) for a master-slave can be repeated multiple times. Each flip-flop after the first aggregates an additional factor \( G(i) = G_{vv}(i) G_{tv}(i) \) and an additional term in the exponent. The general equation for the \( MTBF \) for \( N \) flip-flops becomes:

\[
MTBF(N) = \frac{T \cdot \prod_{i=1}^{N} G(i)}{V_o f_d} \exp \left( \sum_{i=1}^{N} \frac{T}{\tau_{eff}(i)} \right)
\]  

(11)

For the first flip-flop, \( G(1) = G_{tv} G_{vv} \) where the time-to-voltage gain differs from the voltage-to-time gain applicable to succeeding master stages. In general, \( G(i) = G_{vv}(i) G_{tv}(i) \) for \( i > 1 \); \( \tau_{eff}(i) \) is the effective settling time constant for the \( i^{th} \) flip-flop and \( V_o(n) \) is the borderline voltage range for the last flip-flop. Define \( G_{tv}(N) = \prod_{i=1}^{N} G(i) \) as the overall time-to-voltage gain from the D input of the first flip-flop to the Q output of the \( N^{th} \) and last flip-flop. When all flip-flops are identical, we get \( G(i) = G(2) \) for \( i > 1 \) and \( G_{tv}(N) \) is given by:

\[
G_{tv}(N) = G_{tv} G_{vv}(G_{tv} G_{vv})^{N-1} = G(1)(G(2))^{N-1}
\]  

(12)

We can also define an overall effective settling time constant \( \tau_N \) by:

\[
\frac{1}{\tau_N} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\tau_{eff}(i)}
\]  

(13)

The combination of (12) and (13) with (11) gives a familiar bound on the \( MTBF \) of an \( N \)-latch chain:

\[
MTBF(N) \geq G_{tv}(N) T \exp \left( \frac{NT}{\tau_N} \right)
\]  

(14)

For convenience, \( G_{tv} \) and \( V_o \) are often lumped together in a single constant \( T_W(N) = V_o / G_{tv}(N) \), that has dimensions of time. Using this simplification and the clock frequency \( f_c = 1/T \) we obtain:

\[
MTBF(N) \geq \frac{1}{T_W(N) f_d f_c} \exp \left( \frac{NT}{\tau_N} \right)
\]  

(15)
3.3 $N$ Concatenated Flip-Flops used as a Multistage Synchronizer

For the sake of completeness, we will consider the effect of the combination logic ($L_1, L_2$) and flip-flops ($FF_{o1}$ and $FF_{o2}$) on the performance of the multistage synchronizer as presented in Figure 6. The multistage synchronizer shown in Figure 6 is made up of $N+1$ master-slave flip-flops. This multistage synchronizer requires $N+1$ rising clock edges for unsynchronized data to flow through from the input of $FF_1$ to the output of $FF_{N+1}$. Signals should be delivered to the destination synchronous clock domain that satisfy setup and hold times for $FF_{o1}$ and $FF_{o2}$. The two clouds, $L_1$ and $L_2$, represent logic and wire delay that shortens the available settling time of $FF_{N+1}$. Note that the output stage ($FF_{N+1}$) is different from the others because it is likely to have less resolving time. $FF_1$ through $FF_{N+1}$ should be closely coupled with a minimum of inter-stage delay and no logic elements between them. In contrast, the output stage may have long wires to several different clusters of logic before reaching a flip-flop.

Even if all of the stages are well characterized and specified by a standard cell vendor, the SoC designer is faced with the need to incorporate the extrinsic parameters of the application in the calculation of the MTBF. Now (15) the general MTBF formula becomes:

$$MTBF(N + 1) \geq MTBF(N) \geq \frac{1}{t_w(N+1)f_{Dc}} \exp\left(\frac{Nt_{eff}}{t_{N+1}}\right)$$ (16)

where $\max(T_{li})$ denotes the maximum combinational logic delay connected to the last flip-flop. This result has the disadvantage that it is complicated, unfamiliar and mixes extrinsic and intrinsic parameters. A simpler approach that separates the parameters more cleanly and provides a conservative bound on MTBF can be obtained by assuming that the value of $\max(T_{li})$ is unknown and may be large, approaching $T$. This is equivalent to assuming that the $N+1$ stage is ineffective in adding to the settling time and, hence, the input to $FF_{N+1}$ should have valid high ($V_{HH}$) or low ($V_{LL}$) voltages. Now, the lower bound on MTBF for an $N+1$ flip-flop synchronizer becomes:

$$MTBF(N + 1) \geq MTBF(N) \geq \frac{1}{t_w(N+1)f_{Dc}} \exp\left(\frac{NT}{t_{N}}\right)$$ (17)

This bound on $MTBF(N + 1)$ is not as tight as (15), but eliminates the ambiguity associated with the output stage.

3.4 Multistage Synchronizer with Identical Stages

If all $N+1$ flip-flops of a synchronizer standard cell have identical characteristics, (17) can be evaluated in a straightforward manner. In this case, from (9) and (13) it can be shown that $\tau_N = \tau_{eff} = \left(\frac{\alpha}{\tau_M} + \frac{(1-\alpha)}{\tau_S}\right)^{-1}$. Both $\tau_M$ and $\tau_S$ can be found using simulation methods such as in [4][2].

The value of $T_{W}(N)$ can be calculated from $G^*(N)$ and $V_b(N)$ or simulated directly using

$$T_{W}(N) = \frac{V_b(N)}{G^*(N)} = (\delta_+ - \delta_-) \exp\left(\frac{NT}{\tau_{eff}}\right)$$ (18)

Here, $\delta_+$ and $\delta_-$ are those values of data-clock offset that just reach $V_{HH}$ and $V_{LL}$, respectively. For identical stages, $V_b$ is independent of the value of $N$ and we may combine (12) and (18) to obtain a recurrence relation for $T_w(N)$:

$$T_w(N) = \frac{V_b}{G(1)G(2)^{N-1}} = \frac{T_w(N-1)}{G(2)}$$ (19)

If the standard-cell vendor characterizes the synchronizer flip-flops and provides the parameters $T_w(1)$, $T_w(2)$, $\tau_M$ and $\tau_S$, all the terms in (19) are then available to the SoC designer to estimate MTBF. The parameters $N$, $\alpha$, $f_D$ and $f_c$ come from the application. The effective settling time-constant $\tau_{eff}$ can be calculated from (9) given $\alpha$, $\tau_M$ and $\tau_S$. Remember that the number of flip-flops in the synchronizer is $N+1$ and includes an extra one to provide a consistent load for stage $N$. As a result, (17) cleanly
separates extrinsic and intrinsic parameters. This approach disentangles the design of the logic inside the synchronous clock domain from the design of the synchronizer. However, (17) provides only a lower bound on MTBF rather than an estimate. Whether this is an attractive trade-off, compromising accuracy by gaining simplification remains to be seen, but promising indications are presented in the section 4.

3.5 Model Assumptions

As shown in Figure 5, when metastability spans multiple stages, each latch may be metastable for almost half of a clock period. During the first half of the period, the voltage at the master output grows with the settling-time constant \( \tau_M \) and during the last half period, the slave output grows with \( \tau_S \). This exponential behavior is repeated for each succeeding pair of latches throughout a multistage synchronizer, but delayed by the partial period between succeeding pair of latches. Therefore, the voltage at the master output grows with the settling-time constant \( \tau_M \) and during the last half period, the slave output grows with \( \tau_S \). These exponential decays disentangle the design of the logic inside the synchronous clock domain from the design of the synchronizer.

However, (17) provides only a lower bound on \( \tau_M \). As shown in Figure 5, when metastability spans multiple stages, each latch may be metastable for almost half of a clock period. During the first half of the period, the voltage at the master output grows with the settling-time constant \( \tau_M \) and during the last half period, the slave output grows with \( \tau_S \). This exponential behavior is repeated for each succeeding pair of latches throughout a multistage synchronizer, but delayed by the partial period between succeeding pair of latches. Therefore, the voltage at the master output grows with the settling-time constant \( \tau_M \) and during the last half period, the slave output grows with \( \tau_S \). These exponential decays disentangle the design of the logic inside the synchronous clock domain from the design of the synchronizer.

Simplifications about signal edges were made in the derivation of (15). For example, realistic clock edges will have non-zero rise and fall times. There will, however, be a critical time within the edge when a zero-rise time edge will yield similar results. This observation introduces a small variation in timing of the various clock edges, but because of the argument associated with (20) this variation will not change the general character of the result. Similarly, the setup time preceding the falling clock edge at \( \alpha T \) in (2) only changes the multiplicative coefficient. In both these cases, the simulation discovers the modified coefficients \( \tau_M \) and \( \tau_S \) so that (17) gives a tight bound on MTBF.

There will likely be multiple exponentially decaying solutions to the linear differential equations modeling the metastable behavior of the master-slave synchronizer. These transients are not modeled in the above equations, but their effects can be largely removed from simulation by techniques for handling common-mode effects. Since the metastable voltage is reached after those transients effects, the clock period should be constrained to some minimum value in order to enable sufficient time for the metastable condition to develop.

4 Simulations

Figure 7 shows MTBF using formulas of previous publications referenced in Table 1 and the formula derived in this work (17). The calculations are compared with simulations performed using method [4].
**MTBF** is calculated for different clock periods for a four flip-flop synchronizer. All four stages were taken to be identical with a 50% duty cycle and $f_d=200$ Mhz. Simulation values, parameters for calculation and circuit netlists were obtained using a commercial 90nm process. The comparisons include formulas [6], [7] and [13]. Since in those publications there is no differentiation between $\tau_M$ and $\tau_S$, we provide two calculations for both cases. Calculations using the published formulas, but with $\tau_{\text{eff}}$ as in (9), are also shown. The values of $T_{\text{eff}}$ for a single latch were used for all the referenced calculations. Results show a significant improvement in accuracy, by our model, representing the tightest lower bound on the MTBF. Formulas from [11] and [14] may provide a similar accuracy as our model but their formulations are less intuitive so it is hard to make comparisons over a wide range of situations.

Figure 8 shows an example set of calculations and simulations for multiple flip-flop synchronizers and the match between simulation and the developed model. The comparisons include formulas [6], [7] and [13]. Since in those publications there is no differentiation between $\tau_M$ and $\tau_S$, we provide two calculations for both cases. Calculations using the published formulas, but with $\tau_{\text{eff}}$ as in (9), are also shown. The values of $T_{\text{eff}}$ for a single latch were used for all the referenced calculations. Results show a significant improvement in accuracy, by our model, representing the tightest lower bound on the MTBF. Formulas from [11] and [14] may provide a similar accuracy as our model but their formulations are less intuitive so it is hard to make comparisons over a wide range of situations.

**5 CONCLUSIONS**

We developed an expression to accurately estimate a lower bound on the MTBF of multistage synchronizers that enables calculation for an arbitrary number of stages. The formula is based on four extrinsic parameters, $N$, $\alpha$, $f_d$ and $f_c$, and four intrinsic parameters, $T_W(1)$, $T_W(2)$, $\tau_M$ and $\tau_S$. We introduced the concept of $\tau_{\text{eff}}$ and showed the influence of the duty cycle on the resulting MTBF. The resulting formula was compared with previously published formulas. Some formulas compromise accuracy for ease of use, while others provide good estimates but are harder to use. Our formula was demonstrated to be accurate, easy to use and intuitive. Unlike the other methods, ours provides a tight lower bound on the MTBF. For example, Figure 7 shows and expanded view of a typical result for a clock period of 1 ns. The simulation indicates an MTBF that is about a factor of two greater than the slightly more conservative bound calculated according to (17). The formulas from the literature give bounds that are two to five orders of magnitude more conservative.

**6 ACKNOWLEDGEMENT**

The work of Salomon Beer was supported in part by HPI institute for scalable computing. This work was supported in part by the National Science Foundation under Grant No. 0924010. The authors would like to thank the anonymous reviewers for their comments and suggestions that helped improve the quality of this publication.

**7 REFERENCES**


