MTBF Estimation in Coherent Clock Domains

Salomon Beer[1], Ran Ginosar[1], Rostislav (Reuven) Dobkin[2], Yoav Weizman
[1]EE Dept. Technion—Israel Institute of Technology, Haifa, Israel, [2]*vSync Circuits Ltd.*
sbeer@tx.technion.ac.il

***Abstract*— Special synchronizers exist for special clock relations such as mesochronous, multi-synchronous and ratiochronous clocks, while variants of N-flip-flop synchronizers are employed when the communicating clocks are asynchronous. N-flip-flop synchronizers are also used in all special cases, at the cost of longer latency than when using specialized synchronizers. The reliability of N-flip-flop synchronizers is expressed by the standard MTBF formula. This paper describes cases of coherent clocks that suffer of a higher failure rate than predicted by the MTBF formula; that formula assumes uniform distribution of data edges across the sampling clock cycle, but coherent clocking leads to drastically different situations. Coherent clocks are defined as derived from a common source, and phase distributions are discussed. The effect of jitter is analyzed, and a new MTBF expression is developed. An optimal condition for maximizing MTBF and a circuit that can adaptively achieve that optimum are described. We show a case study of metastability failure in a real 40nm circuit and describe guidelines used to increase its MTBF based on the rules derived in the paper.**

Keywords- *Synchronization, metastability, mean time between failures (MTBF), coherent clocks.*

1 Introduction

Recently, a SoC product exhibited an alarmingly high rate of random failures in operation. Analysis showed that the problem was located in a clock domain crossing based on a two flip-flop synchronizer. While such synchronizers are designed to bridge asynchronous clock domains, it turned out that the domains in question were coherent, resulting in an increased failure rate.

The usual classification [1][2][4] sorts the relationship between two clocks based on their frequency and phase relations, as in the upper part of Figure 1. No frequency and phase relationship is assumed for two asynchronous clocks, and various relations exist in the loosely synchronous class. That class is further divided into mesochronous, plesiocronous and heterochronous groups. The latter group is further sub-divided into ratiochronous and non-ratiochronous [5][6] clocks. We employ a different classification, based on clock sources, as shown in the lower part of Figure 1. Clocks are non-coherent when they are sourced from different references and coherent when they share a common reference clock. The coherent case is further divided into two subcases depending on the nature of their phase distribution, uniform and non-uniform. The dotted circle groups the cases where the phase distribution is uniform, whether coherent or not. Note that some cases of the loosely synchronous clocks may be either coherent or non-coherent, and hence it is impossible to unify the two different classifications. It is widely believed that loosely synchronous clock domains may be synchronized using either a special purpose synchronizer designed for each of the special cases(e.g., [7]-[11]), or using a brute-force N-flip-flop synchronizer of the type designed for asynchronous domains [2][3][4][19].

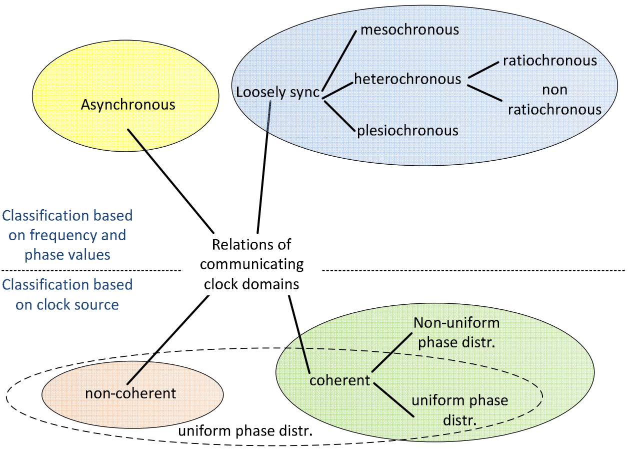

Figure 1. Two different classifications for clock relation

In that latter case, the reliability of the synchronizer is given by the estimate of mean time between failures ($MTBF$):

$$MTBF = \frac{e^{S/\tau}}{T_W \cdot f_C \cdot f_D} \qquad (1)$$

where f_C, f_D, S denote the frequency of the clock, the rate of the incoming data signal and the settling time allowed for synchronization between the clock domains, and τ and T_W are the metastability resolution time constant and its window of vulnerability. Significant research has focused on the improvement and enhancement of such synchronizers [12][13][14]. However, we have realized that in certain cases of coherent clocks that expression does not apply.

Deep inside (1) lays the assumption that the probability distribution of data edges along the sampling clock period is uniform [2][15]. However, we show that uniform distribution cannot be assumed in coherent clock domains. Rather, the common clock source leads to particular non-uniform phase relations that may result in significantly higher failure rates than predicted by (1).

The paper is organized as follows. Section 2 defines coherent clocks and discusses the resulting phase relations. Section 3 describes jitter noise and its influence on clock phases. In Section 4 we develop a formula for $MTBF$ in a general coherent clock case and an optimality condition for minimum failure rate. In section 5 we show the conditions for achieving that minimum and explain why previous publications [7]-[11] on adaptive synchronization do not provide such optimality. Section 6 presents the case study of a synchronization failure in a Soc, as discussed at the beginning of this introduction, showing solutions to achieve the optimal condition and section 7 concludes the work.

2 Coherent Clocks

Synchronization in multiple-clock domain SoC can be sorted into two major categories, coherent and non-coherent clocking. The coherent clocks scenario is illustrated in Figure 2. Two clock domains are fed from two different PLLs that are referenced from a common source and apply rational frequency multipliers M$_1$,M$_2$. The clock frequencies of domains 1 and 2 are f_1 and f_2 respectively. A data signal sourced in domain 1 is sampled by a flip-flop in domain 2. This is the case when clock domains are referenced from a single oscillator or crystal on board that provides reference to all domains. In the general case, no assumption is made on the values of f_1 and, f_2 , and every ratio is permitted according to the programmed values of the PLL multipliers. This case is similar to Globally-Ratiochronous, Locally-Synchronous (GRLS) in [6].

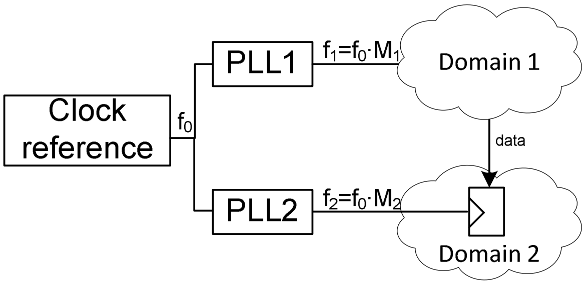

Figure 2. Coherent clocks

The non-coherent scenario is illustrated in Figure 3, and corresponds to the case where the communicating clock domains are sourced from different references. This is the case when more than one oscillator is present in the circuit, or when synchronizing asynchronous inputs into the system. Both coherent and non-coherent cases may be present in large SoCs.

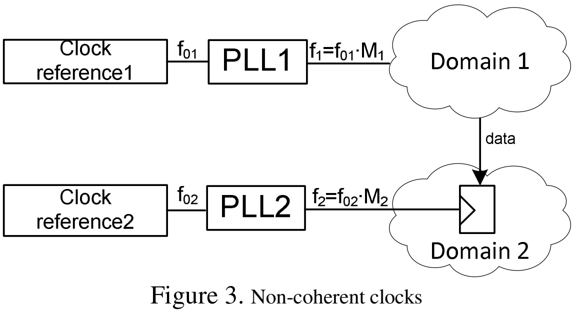

Figure 3. Non-coherent clocks

In the following we deal mostly with the coherent clock scenario; non-coherent clocks are discussed briefly at the end of Section 3. In Figure 2, data is generated at rate f_1. The aim is to analyze the distribution of the phase difference between the data leading edge and its sampling clock in domain 2. We denote by $\varphi(n)$ the time difference in the n^{th} cycle of clock f_1 (this time difference is henceforth expressed as a phase difference). Assuming $f_2 > f_1$, the phase is bounded by $0 \leq \varphi(n) \leq T_2$. Figure 4 describes this scenario; the leading edge of data is represented by the rising edge of clock 1.

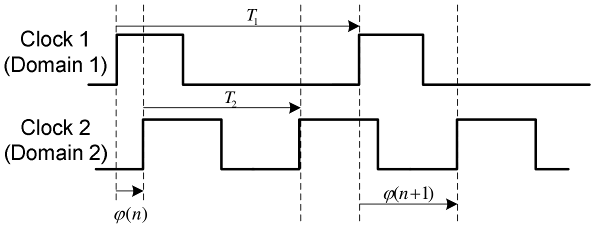

Figure 4. Relative phases of two clocks

Because both clocks are derived from a common reference, they are rational,

$$\eta = \frac{f_2}{f_1} = \frac{f_0 M_2}{f_0 M_1} = \lfloor \eta \rfloor + \frac{P}{Q} \tag{2}$$

Where $P = M_2/\gcd(M_1, M_2)$ and $Q = M_1/\gcd(M_1, M_2)$. Following (2) and the waveform diagram of Figure 4, an equation describing the evolution of phase for cycle n can be derived [16].

$$\varphi(n) + K \cdot T_2 = \varphi(n+1) + \eta \cdot T_2 \tag{3}$$

where K can take only two possible values, $K = \lfloor \eta \rfloor$ or $K = \lfloor \eta \rfloor + 1$. Equation (3) has been studied in the context of communication systems [17] and the solution is given by:

$$\varphi(n) = \varphi(0) - n \cdot \rho \cdot T_2 - \left\lfloor \frac{\varphi(0)}{T_2} - n \cdot \rho \right\rfloor \cdot T_2 \tag{4}$$

where $\rho = \eta - \lfloor \eta \rfloor = \frac{P}{Q}$ and $\varphi(0)$ is the phase at time zero. An interesting property of (4) is that $\varphi(n+Q) = \varphi(n)$, which means that $\varphi(n)$ is periodic with period Q and $\varphi(n)$ can take at most Q different values.

$$\begin{aligned} \varphi(n+Q) = {} & \varphi(0) - (n+Q) \cdot \rho \cdot T_2 - \left\lfloor \frac{\varphi(0)}{T_2} - (n+Q)\rho \right\rfloor \cdot T_2 \\ = {} & \varphi(0) - (n+Q) \cdot \frac{P}{Q} \cdot T_2 - \left\lfloor \frac{\varphi(0)}{T_2} - (n+Q)\frac{P}{Q} \right\rfloor \cdot T_2 \\ = {} & \varphi(0) - n \cdot \rho \cdot T_2 - \lfloor \varphi(0) - n\rho \rfloor \cdot T_2 = \varphi(n) \end{aligned} \tag{5}$$

An exhaustive analysis [16] of the solution (4) shows that $\varphi(n)$ is composed of P monotonically decreasing sub-sequences as shown in Figure 8. The possible values of $\varphi(n)$ are uniformly distributed in the interval $[0, T_2]$ and the distance between two aligned values of $\varphi(n)$ is then T_2/Q. Figure 5 shows $\varphi(n)$ for a synchronization scenario where f_1 is 100Mhz and f_2 is 150Mhz. In this case P=1,Q=2 and as expected the phase between the two signals takes only two values. Another example in shown in Figure 6 and its histogram in Figure 7, where f_1=125Mhz and f_2=150Mhz. The frequency ratio is $f_2/f_1 = 1 + 1/5$, and the phase takes only five different discrete values.

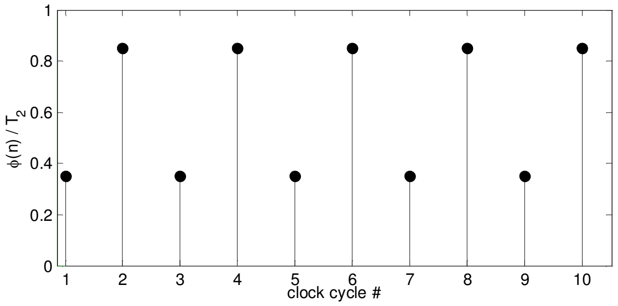

Figure 5. Evolution of $\varphi(n)$ over time for f_1=100Mhz and f_2 = 150Mhz

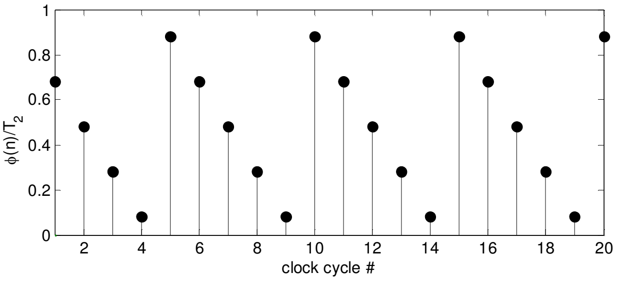

Figure 6. Evolution of $\varphi(n)$ over time for f_1=125Mhz and f_2 = 150Mhz

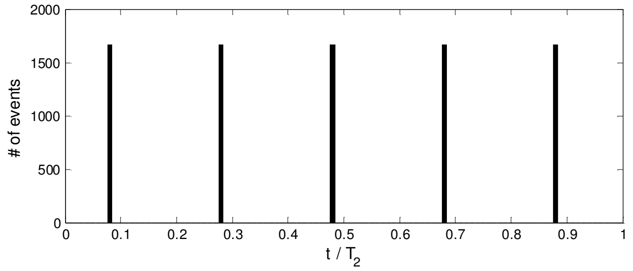

Figure 7. Histogram of $\varphi(n)$ for f_1=125Mhz and f_2 = 150Mhz, P=1 Q=5

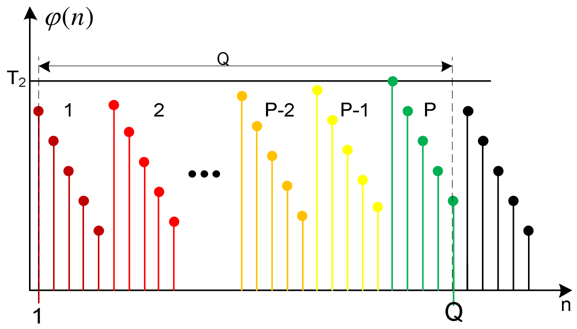

Figure 8. $\boldsymbol{\varphi(n)}$ solution for rational coherent frequencies

Based on the derived results a general expression for the probability density function (pdf) of the phase $\varphi(n)$ can be obtained:

$$pdf_{\varphi(n)}(t) = \frac{1}{Q}\sum_{i=1}^{Q} \delta\big(t - \varphi(i)\big)\,, \qquad t\epsilon[0, T_2] \tag{6}$$

where $\delta(x)$ is the Dirichlet delta function. Figure 9 shows a diagram of the pdf for the phases.

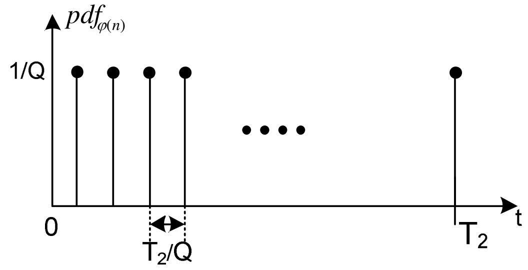

Figure 9. Probability density function diagram of $\boldsymbol{\varphi(n)}$

2.1 *Small perturbations in clock frequencies*

In the previous examples f_2/f_1 took perfect rational numbers with low Q values, corresponding to only a few possible values of the phase $\varphi(n)$. What happens if a small perturbation appears in one of the clocks? To answer this question, we denote the number of different possible phases for frequencies f_1 and f_2 as $\omega(f_1, f_2)$. As seen in the previous sub-section, for $f_2/f_1 = \lfloor\eta\rfloor + P/Q$, $\omega(f_1, f_2) = Q$. A special property related to $\omega(f_1, f_2)$ is that for a small frequency perturbation, say $(f_1, f_2 + \varepsilon)$, the value $\omega(f_1, f_2 + \varepsilon)$ may become unbounded. A formal proof of this discontinuity of $\omega(f_1, f_2)$ requires precise definitions and is beyond the scope of this paper. Instead, a general guideline about this claim is presented. First, we denote the deviation from f_2 as a fraction of f_2 , meaning $\varepsilon = f_2 \cdot \theta_N/\theta_D$, and for small perturbations $\theta_D \gg \theta_N$. Then, the number of possible phases for the perturbed frequency values is given by $\omega(f_1, f_2 + \varepsilon) = \tilde{Q}$, where $\frac{f_2+\varepsilon}{f_1} = \lfloor\tilde{\eta}\rfloor + \frac{\tilde{P}}{\tilde{Q}}$. Substituting, we obtain the expression:

$$\lfloor\tilde{\eta}\rfloor + \frac{\tilde{P}}{\tilde{Q}} = \frac{f_2 + \frac{\theta_N}{\theta_D}\cdot f_2}{f_1} = \lfloor\eta\rfloor + \frac{P\cdot\theta_D + \lfloor\eta\rfloor\cdot\theta_N\cdot Q + P\cdot\theta_N}{Q\cdot\theta_D} \tag{7}$$

From (7), $\tilde{Q}$ can be at most $Q\cdot\theta_D$ and since $\theta_D$is a large number, the number of possible phases $\omega(f_1, f_2 + \varepsilon) = Q\cdot\theta_D$ is very large. In some cases, the numerator and denominator in (7) may have common divisors, lowering the value of ω. In summary, while for certain frequencies f_1, f_2 the number of possible phases can take a limited number of values (Q), for a small perturbation of those frequencies the number of possible phases may drastically increase.

To show the above argument, we consider a case where $f_1 = 125Mhz$ and $f_2 = 151.5Mhz$, meaning $\varepsilon = 1.5\,Mhz$ and $\frac{\theta_N}{\theta_D} = 0.01$, representing a 1% deviation from $f_2 = 150Mhz$. Figure 10 and Figure 11 show $\varphi(n)$ and its histogram. The number of possible phases increased from 5 to 250 possible phases. The histogram in Figure 11 resembles a continuous uniform distribution, since it is composed of a large number of delta functions.

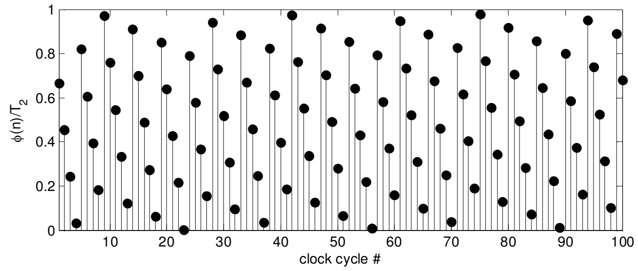

Figure 10. $\varphi(n)$ for f_1=125Mhz and f_2 = 151.5Mhz

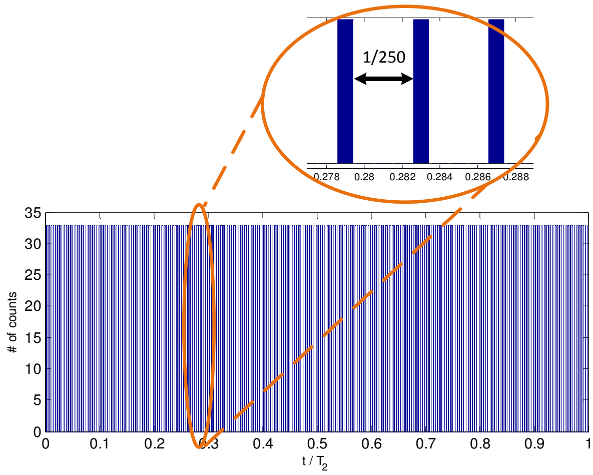

Figure 11. Histogram of $\varphi(n)$ for f_1=125Mhz and f_2 = 151.5Mhz, P=53 Q=250. A small segment is magnified for clarity.

3 Clock Phase Probability Distribution

The preceding analysis ignores noise. In periodic electronic signals, noise manifests as phase jitter. To understand the effect of jitter on the values of $\varphi(n)$, we assume the noise is independent, time invariant and additive [18]. Then,

$$\varphi_N(n) = \varphi(n) + \nu(n) \qquad (8)$$

where $\varphi_N(n)$ describes the jittered phase at cycle n, $\nu(n)$ is the jitter component that is assumed to have normal distribution $N(0,\sigma^2)$, and $\varphi(n)$ are the ideal phase values as described in the previous section. Figure 12 and Figure 13 show the effect of noise on the phase positions for f_1=125Mhz and f_2 = 150Mhz case. As expected, instead of delta like phase positions, Gaussian like distributions in each of the peaks are obtained. Figure 14 and Figure 15 show a similar example for the case of a slight deviation from the desired frequencies. Since the number of possible phase positions increases drastically, the final result is almost a continuous uniform distribution through all possible phases.

In the *non-coherent* scenario Figure 3, the ratio of the two clock frequencies cannot in general be expressed as a rational number. This is true even when the two reference clocks are specified to the same nominal frequency. Rather, this ratio is modeled as a rational number plus a small perturbation. Hence, based on the analysis of small perturbations explained above, the relative phases span a wide range in a manner close to uniform distribution. This situation persists even when adding noise.

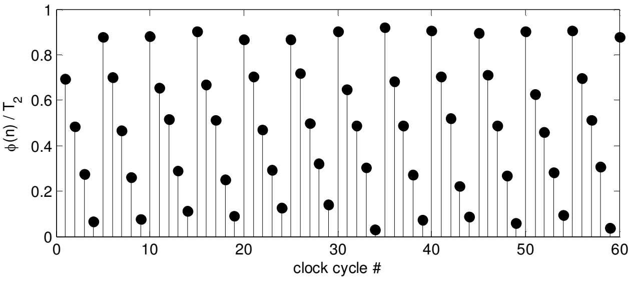

Figure 12. $\varphi_N(n)$ for f_1=125Mhz and f_2 = 150Mhz with jitter

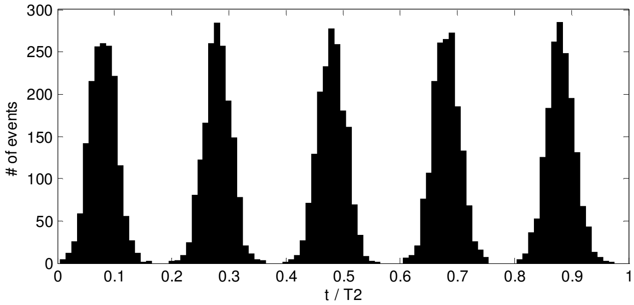

Figure 13. Histogram of $\varphi_N(n)$ for f_1=125Mhz and f_2 = 150 Mhz, P=1 Q=5 $\sigma = 0.06T_2$

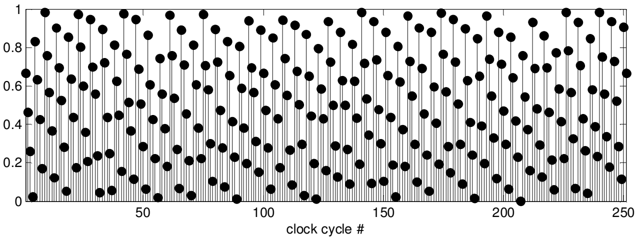

Figure 14. $\varphi_N(n)$ for f_1=125Mhz and f_2 = 151.5Mhz

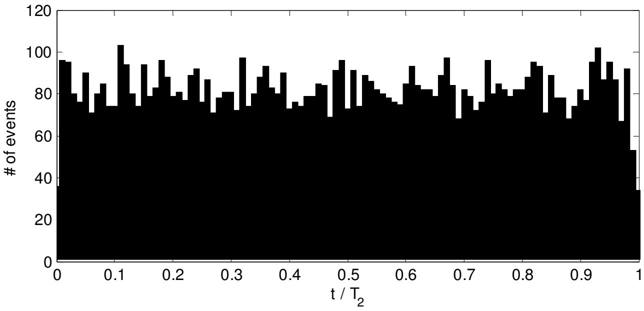

Figure 15. Histogram of $\varphi_N(n)$ for f_1=125Mhz and f_2 = 151.5Mhz, P=53 Q=250

So far we have shown that the different clock scenarios can be classified based on their clock reference. In the non-coherent case the phase is distributed uniformly, while in the coherent case the phase distribution may become non-uniform depending on the clock frequencies. When the phase is non-uniformly distributed, $MTBF$ calculated using (1) is invalid; a new expression for the coherent non-uniform case follows in the next section.

4 Coherent Clock MTBF

A failure in a synchronizer appears when the data-clock separation is inside the metastability window of vulnerability. Then synchronizer failure probability can be expressed as

$$P(failure) = P(|\varphi_N(n) - \delta_0| < \frac{\delta_W}{2}) \qquad (9)$$

where δ_0 is the theoretical phase separation that causes the synchronizer output to settle at the metastability voltage (V_m) [15]. The parameter δ_W is the metastability window around δ_0 such that for $\varphi_N(n)$ values outside the interval $[\delta_0 - \frac{\delta_W}{2}, \delta_0 + \frac{\delta_W}{2}]$, the voltage at the output of the synchronizer takes defined valid values within bounded time and there is no risk of further metastability propagation. However, if $\varphi_N(n)$ lies inside the interval, the synchronizer output is delayed generating intermediate voltages at its output at the system sampling time which may propagate metastability to the synchronous domain and lead to a failure (Figure 16). δ_W is assumed symmetrical for the ease of the derivation while in real circuits it may be non-symmetric around δ_0.

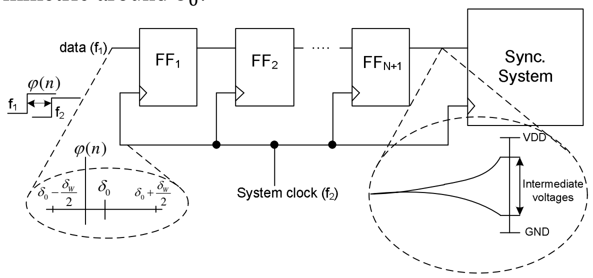

Figure 16. Synchronization diagram

Following (6) and (8), and since the probability density function of the sum of two independent random variables is the convolution of their separate density functions, the *pdf* of $\varphi_N(n)$ can be written as

$$pdf_{\varphi_N(n)}(x) = \frac{1}{Q}\frac{1}{2\pi\sqrt{\sigma}}\sum_{i=1}^{Q} e^{-\left(\frac{t-\varphi(i)}{\sigma}\right)^2}, t\epsilon[0, T_2] \qquad (10)$$

The value of σ is the standard deviation of the jitter noise in the circuit. The probability density can be regarded as cyclic with cycle T_2. The resulting pdf function takes a form similar to the diagrams of Figure 17.

From (10) and Figure 17, we identify two different scenarios. When $T_2 > 2Q\sigma$, (10) represents a non-uniform distribution as in Figure 17(a), having maxima and minima similar to the example of Figure 13. This happens because the distance between the ideal phase positions (T_2/Q) is larger than the standard deviation (σ) of the noise and the maxima are well separated. When $T_2 < 2Q\sigma$, the summation in (10) produces a mixture that can be approximated by a continuous uniform distribution, as in Figure 17(b). This is because the Gaussian mean locations $(\varphi(i))$ are uniformly distributed through the clock period and the distance between phase positions is shorter than the standard deviation of the noise. An alternative analysis in the Fourier domain yields a similar criterion for the uniformity of the overall distribution.

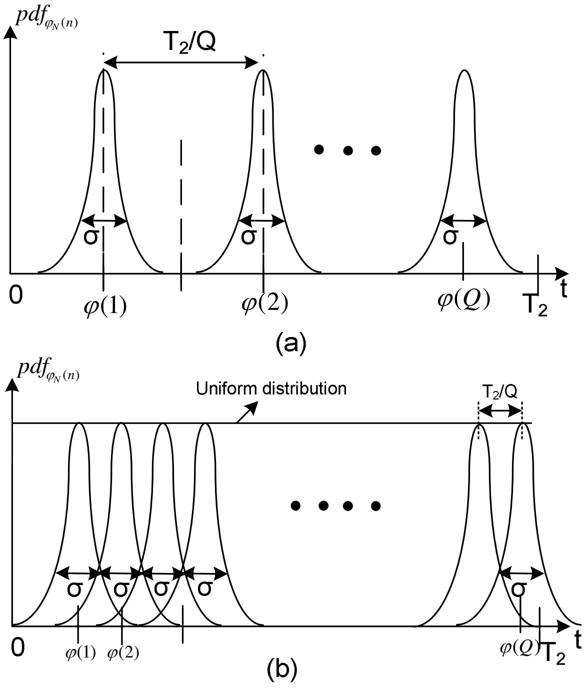

Figure 17. Probability density function diagram of $\varphi_N(n)$, (a) for $T_2 > 2Q\sigma$. (b) $T_2 < 2Q\sigma$

Using (10) the failure probability can be re-written as:

$$P(|\varphi_N(n) - \delta_0| < \frac{\delta_W}{2}) = \frac{1}{Q}\frac{1}{2\pi\sqrt{\sigma}}\int_{\delta_0-\frac{\delta_W}{2}}^{\delta_0+\frac{\delta_W}{2}} \left(\sum_{i=1}^{Q} e^{-\left(\frac{x-\varphi(i)}{\sigma}\right)^2}\right) dt \qquad (11)$$

When $T_2 < 2Q\sigma$, the $pdf_{\varphi_N(n)}$ is approximately constant, and $P\left(|\varphi_N(n) - \delta_0| < \frac{\delta_W}{2}\right) = \frac{\delta_W}{T_2}$ which is the usual result for uniform phase distribution [15].

Assuming the data rate is given by f_1, a new general $MTBF$ expression for the coherent clocks scenario can be derived from (11) :

$$MTBF = \frac{1}{failure\,rate} = \frac{1}{P(|\varphi_N(n) - \delta_0| < \frac{\delta_W}{2}) \cdot f_1} \qquad (12)$$

Evidently, coherent clocking may lead to a different MTBF than expression (1).

The maximum possible $MTBF$ ratio is given when δ_0 lies on a peak or in a valley of the probability distribution function. In those cases, the $MTBF$ ratio is given by:

$$\frac{MTBF_v}{MTBF_p} = \frac{MTBF_{max}}{MTBF_{min}} \approx \frac{exp\left(-\frac{\varphi(i)-\varphi(i)}{\sigma}\right)^2}{exp\left(-\frac{\left(\varphi(i)+\frac{T_2}{2Q}\right)-\varphi(i)}{\sigma}\right)^2} = e^{\left(\frac{T_2}{2Q\sigma}\right)^2} \qquad (13)$$

In common cases, jitter represents a few percent of the clock period. Taking $\sigma/T_2 = 0.02$ (2%) and $Q = 3$, the $MTBF$ ratio becomes 4160, meaning $MTBF$ may increase or decrease by 4-5 orders of magnitude. This $MTBF$ variation should be added to other design margins by increasing the settling time by 9τ ($\ln(4160) \approx 9$), which in modern technologies (especially LP) can add up to 0.5-1 nsec latency. A similar scenario is shown in [20] by means of a special feedback setup that creates metastable events in almost every cycle. When the jitter is extremely low, the MTBF ratio becomes very high (e.g. for 0.5% jitter the ratio is almost 10^{16}), which is a considerable improvement in MTBF.

In most cases the MTBF uncertainty caused by coherent clocks should be compensated by an additional settling time margin. Those margins in synchronizer design should be added to other Process technology, temperature and supply voltage margins (PVT).

5 MAXIMIZING MTBF

Many synchronizers have been proposed for different types of coherent clock relations, such as mesochronous, multi-synchronous, plesiochronous and periodic clocks. However, in this section we consider maximizing the MTBF or N-flip-flop synchronizers, when employed between coherent clock domains.

Because (10) can be non-uniform as described previously, we aim at optimizing the synchronization setup in order to maximize MTBF. Since δ_0 and δ_W are intrinsic parameters of the flip-flops and σ is related to the jitter of the clock network (basically the jitter of the reference clock from which both f_1 and f_2 are derived), we focus our optimization on the phases $\varphi(i)$. The absolute phases are a function of interconnect and internal delays, which determine the value of $\varphi_N(0)$, while the relative phase is independent of any delay and is given by T_2/Q. Internal flip-flop delays depend on the circuit design and hence the only available parameter in a system level perspective is the interconnect delay that affects $\varphi(i)$ by an overall offset.

To find the optimum value of $\varphi(i)$ that yields maximum MTBF, one should solve equation (14):

$$\varphi(i) = \underset{\varphi(i)}{\operatorname{argmax}} MTBF \qquad (14)$$

Since the MTBF function is monotonic, it follows

$$\varphi(i)_{opt} = \underset{\varphi(i)}{\operatorname{argmin}} \left\{ \int_{\delta_0-\frac{\delta_W}{2}}^{\delta_0+\frac{\delta_W}{2}} \left(\sum_{i=1}^{Q} e^{-\left(\frac{x-\varphi(i)}{\sigma}\right)^2} \right) dt \right\} \qquad (15)$$

Since (15) does not have analytical solution, an approximation is given by:

$$\varphi(i)_{opt} = \delta_0 - \frac{\delta_W}{2} - \frac{\overbrace{(\varphi(i+1)-\varphi(i))}^{T_2/2Q}}{2} \approx \delta_0 - \frac{T_2}{2Q} \qquad (16)$$

A graphical representation of the solution is shown in Figure 18. The analytical solution matches the intuitive approach that the interconnect delay is to be adjusted so that the point δ_0 lies between any two peaks $\varphi(i)$ values.

It is possible to build a circuit that produces the optimal MTBF condition derived in (16). Since δ_0 is usually not known to the system designer, a method for adaptive delay learning is implemented. Previous works on adaptive synchronization [7]-[11] do not take δ_0 into consideration and consequently may be unable to achieve the maximum $MTBF$.

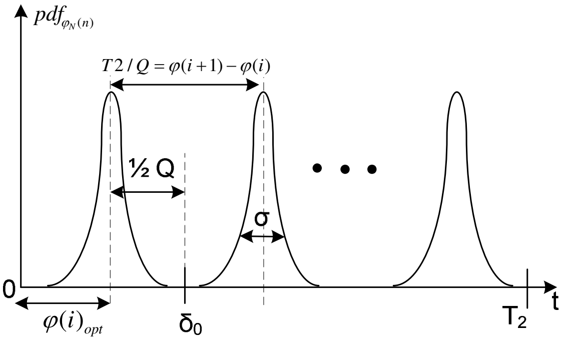

Figure 18. Graphical representation of solution of (15)

The principle of the adaptive delay unit is shown in Figure 19 and consists of a variable delay and delay control block that are independent of the synchronizer. The delay control receives both f_1, f_2 and the output of the first flip-flop in the synchronizer and generates a control signal (set) that sets the delay value. The output of the first synchronizer stage is critical in order to generate the condition shown in (16). Once the delay is found, the control unit locks the value. This procedure is triggered after a reset of the clock domains and the delay is kept locked until any of the frequencies is changed.

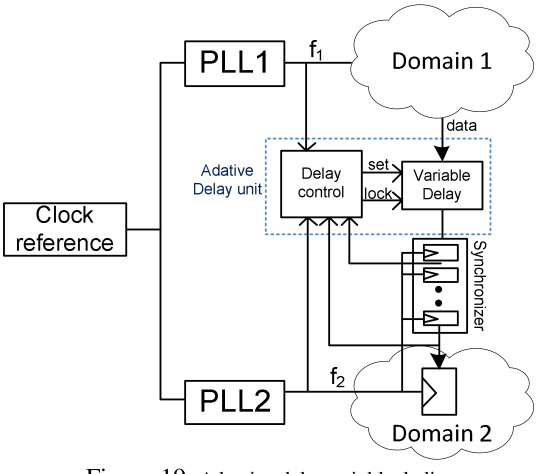

Figure 19. Adaptive delay unit block diagram

6 Case Study

In this section we present a real circuit that *a-posteriori* was found to present metastability failures. The circuit was part of a commercial SoC in a 40nm technology. This presentation aims to achieve two goals: first, to demonstrate that phase distribution may be non-uniform in coherent clock circuits as shown in previous sections; second, to describe techniques useful for detection and analysis of random metastability failures.

The relevant portion of the circuit is shown in Figure 20.

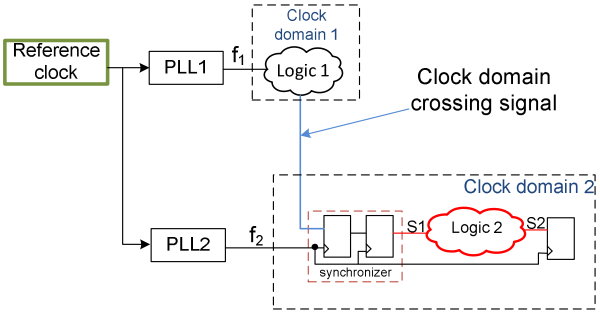

Figure 20. Block diagram of circuit showing failing circuitry

In order to locate the failure, Infra-red emission analysis (IREM) was used. It identified an area of the SoC that exhibited irregular emissions correlated to the failure event. Figure 22 shows the IREM image during normal system operation and just prior to failure. In normal operation, only one emission spot was visualized. Prior to a failure, additional emission spots were observed. Multiple signals in the vicinity of the culprit location were examined, by adding FIB micro-probes. Figure 21 shows the synchronizing clocks and the waveform at the output of the synchronizer (signal S1), with an unexpectedly-short pulse, caused by a late output transition generated by metastability. Logically, this event was determined to cause the failure.

Since the system employed coherent clocks, we studied the phase relation of the micro-probed clock and the data feeding into the suspected synchronizer. Figure 23, Figure 24 and Figure 25 show waveforms and histograms of the phase distribution between f_1 and f_2 for different configurations, captured by an oscilloscope. Clock waveforms are shown in purple, phase histogram is shown in blue, a zoomed version of the phase histogram is shown in green and the measured period for f_2 is shown in yellow. In Figure 23 the ratio $f_2/f_1 = 8 + 1/3$, giving three peaks in the histogram. In Figure 24 the ratio $f_2/f_1 = 8 + 1/5$, giving 5 peaks, and in Figure 25, $f_2/f_1 = 8 + 1/2$, giving 2 peaks. The value of δ_0 was obtained from simulations and static timing analysis and its position is marked in each of the graphs.

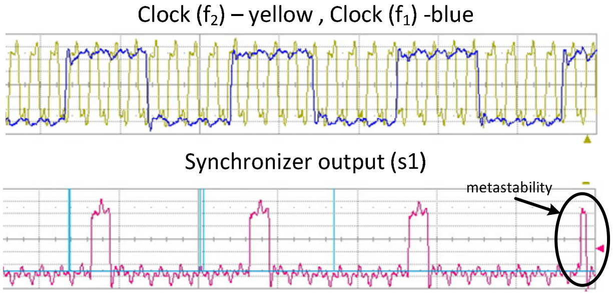

Figure 21. Oscilloscope waveform at the output of the synchronizer (S1)

Using the value of δ_0 and the phase histogram, we calculated the probability of failure, by the ratio of the events in a window around δ_0 divided by the overall event number in the histogram. Table 1 shows the result of the failure probability for different values of the f_2/f_1 ratio. The failure probability changes by very large factors. We then validated these findings by measuring failure probabilities of the SoC for each of the tabulated ratios. Finally, we directed the SoC user to use only the ratios that are highlighted in the table, since they lead to significantly reduced failure probabilities.

We note that the proposed solution did not fix the problem completely, but increased the $MTBF$ by two orders of magnitude, which resulted in an acceptable solution for the specific application.

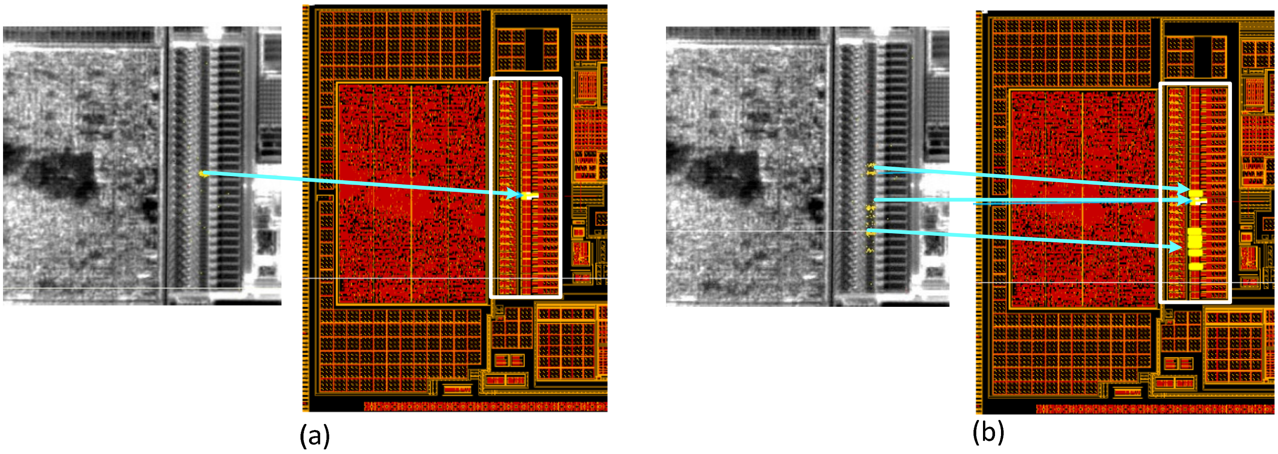

Figure 22. IREM and layout mapped IREM (a) during normal operation. (b) During system failure

In the future, a similar SoC may employ a circuit like in Figure 19 to dynamically adjust δ_0 and achieve an even better improvement in $MTBF$.

The synchronizer in Figure 20 was poorly designed and the case is presented here to illustrate the coherent clock phase distribution and not as a method to solve metastability issues.

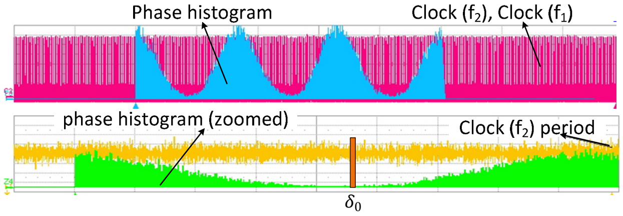

Figure 23. Phase histogram for $f_2/f_1 = 8 + 1/3$

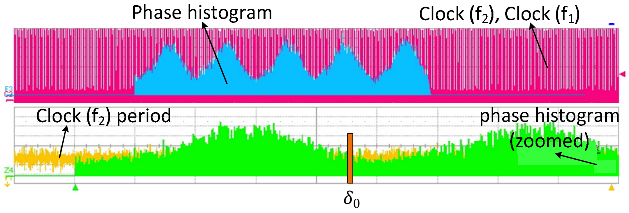

Figure 24. Phase histogram for $f_2/f_1 = 8 + 1/5$

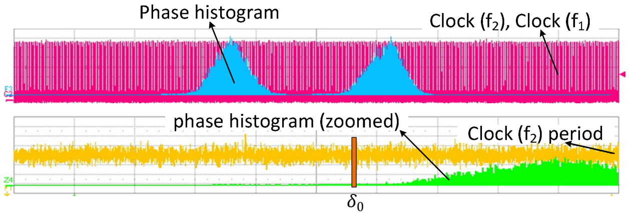

Figure 25. Phase histogram for $f_2/f_1 = 8 + 1/2$

Table 1. Failure probability for different f_2/f_1 ratios

f2/f1 Ratio	Event Ratio	Probability
8+1/3	15/63800	0.0002351
8+1/4	80/77300	0.0010349
8+2/3	15/55700	0.0002693
8+1/2	5/27600	0.0001812
8+3/4	5/23500	0.0002128
8+1/5	50/62400	0.0008013
8+2/5	50/48900	0.0010225
8+3/5	50/57100	0.0008757
8+4/5	50/50900	0.0009823
9+2/3	30/51400	0.0005837
9+4/5	55/52800	0.0010417
9+3/5	60/54200	0.0011070
9+3/4	55/51800	0.0010618
9+1/2	50/53300	0.0009381

7 Conclusions

We have proposed a new classification of CDC synchronization based on the source of the clock references involved. Coherent and non-coherent clock scenarios are introduced. In the non-coherent clock scenario the clock phase distribution is shown to be uniform as assumed in previous publications. In contrast, coherent clock synchronization is shown to present non-uniform phase distribution in some cases depending on clock frequencies. A condition for non-uniformity versus uniformity in coherent clocks is developed. A new formula for $MTBF$ in the general coherent clock scenario is developed and an expression for optimum $MTBF$ is found. A general block diagram of an adaptive synchronization scheme that can maximizes $MTBF$ is proposed. A real case of synchronization failure in a coherent clocking SoC is presented, demonstrating measured non-uniform phase distribution and also illustrating how random metastability failures can be detected and localized in real chips.

8 Acknowledgement

The work of Salomon Beer was supported in part by HPI Institute for scalable computing. The authors would like to thank the anonymous reviewers for their wise comments that helped improving the quality of this publication.

References

[1] D.G. Messerschmitt, "Synchronization in Digital System Design," IEEE J. Selected Areas in Communications, 8(8):1404-1419,1990.

[2] P. Teehan, M. Greenstreet, G. Lemieux, "A Survey and Taxonomy of GALS Design Styles," IEEE Design & Test of Computers, 24(5):418-428, 2007.

[3] J. M. Chabloz and A. Hemani, "Distributed DVFS using rationally-related frequencies and discrete voltage levels", ISLPED '10.

[4] W.J. Dally and J.W. Poulton. *Digital systems engineering*. Cambridge university press, 1998.

[5] M.W. Heath, W.P. Burleson, and I.G. Harris, "Synchrotokens:A Deterministic GALS Methodology for Chip-Level Debug and Test," IEEE Trans. Computers, 54(12):1532-1546, 2005.

[6] J. M. Chabloz and A. Hemani, "A Flexible Interface for Rationally Related Frequencies," ICCD, pp. 109-116, 2009.

[7] U. Frank, T. Kapschitz and R. Ginosar, "A Predictive Synchronizer for Periodic Clock Domains," Formal Methods in System Design, 28(2):171-186, 2004.

[8] R. Kol and R. Ginosar, "Adaptive Synchronization", ICCD, 1998.

[9] W.K. Stewart, S.Ward, "A solution to a special case of Synchronization Problem," IEEE Trans. Comp., 37(1):123-125, 1988.

[10] L.F.G. Sarmenta, G.A. Pratt, S.A. Ward, "Rational clocking," ICCD, 271-278, 1995.

[11] W.J. Dally, S.G. Tell, "The Even/Odd Synchronizer: A Fast, All-Digital, Periodic Synchronizer," ASYNC, pp. 75-84, 2010.

[12] D.J. Kinniment, A. Bystrov and A.V. Yakovlev, "Synchronization circuit performance," JSSC, 37(2):202-209, 2002.

[13] I.W. Jones, S. Yang and M. Greenstreet, "Synchronizer Behavior and Analysis," ASYNC, pp. 117-126, 2009.

[14] J. Zhou, D. Kinniment , G. Russell and A. Yakovlev, "A robust synchronizer," Symposium on Emerging VLSI Technologies, 2006.

[15] D. Kinniment, Synchronization and Arbitration in Digital Systems, Wiley, 2007.

[16] A. Cantoni, J. Walker and T. Tomlin, "Characterization of a Flip-Flop Metastability Measurement System," IEEE Trans. Circuits and Systems 54(5):1032-1040, 2007.

[17] R. Lau and P. Fleischer, "Synchronous techniques for timingrecovery in BISDN" IEEE Trans. Comm. 43(4):1810–1818, 1995.

[18] R. Navid, T. Lee and R. Dutton, "An Analytical Formulation of Phase Noise of Signals With Gaussian-Distributed Jitter," IEEE Trans. Circuits and Systems, 52:249-153, 2005.

[19] R. Ginosar, "Fourteen ways to fool your synchronizer," ASYNC, pp. 89-96, 2003.

[20] D.J Kinniment,. C.E. Dike, K. Heron, G. Russell; A.V. Yakovlev, "Measuring Deep Metastability and Its Effect on Synchronizer Performance," TVLSI, 15(9):1028-1039, 2007.