

Characterizing Metastability

Practical Measurement Techniques to accurately determine "device dependent coefficients" used to predict synchronizer MTBF.

by
Clark Foley

Tektronix, Inc

Abstract

Determining metastability characteristics is challenging. Devising reliable and repeatable experiments and procedures requires time, patience, care and knowledge. This discussion presents practical measurement techniques to accurately determine the Resolving Time Constant (τ) and Metastability Window (W). Also included is a method for observing the metastability failure rate at a designated time following the Clock. By converting this failure rate to observed MTBF (Mean Time Between Failure), a comparison is made to a predicted MTBF.

Introduction

Determining metastability characteristics is challenging. Devising reliable and repeatable experiments and procedures requires time, patience, care and knowledge. Although a wealth of information has been published regarding synchronization and metastability, such as the ever popular and excellent "late transition detector" method presented by Peter Stoll (ref [1]), all of the techniques have proven difficult to reliably recreate and use.

A few years ago, Martin Bolton (ref [2]) made the observation that despite the large number of measurements which have been reported, there is still a need for standard test methods. A number of results cannot be compared because of different excitation and recording methods. This still appears to be the case. The methods presented here are not likely to be the final solution but perhaps this discussion will serve as an advancement toward standardization.

This discussion presents practical measurement techniques to accurately determine the Resolving Time

Constant (τ) and Metastability Window (W). Also included is a method for observing the metastability failure rate at a designated time following the Clock. By converting this failure rate to observed MTBF (Mean Time Between Failure), a comparison is made to a predicted MTBF.

The methods offered here can be easily applied to today's fastest logic. Other than the device under test, no additional external active circuitry is required. All of the stimulus, acquisition and power is provided by purchased standard test equipment. No custom or home-grown gadgetry is needed. A personal computer can be used to control the instruments via IEEE-488. In this analysis, some "hand-crafted" software was created to easily and consistently conduct tests and compute and plot results.

For brevity, these methods are applied to characterizing a single flip-flop. From this example, these techniques can be easily applied to other synchronizers.

Analysis of a Single-Stage Synchronizer

Assume that you are faced with this problem. For a single-stage synchronizer (D Flip-Flop) shown in figure 1, predict the MTBF for the simple synchronizer to reach a legal logic level at some time, t , after the clock edge, where t is one set-up time prior to the next clock edge, an illegal logic level is between the minimum valid "high" output voltage (V_{OH_Min}) and the maximum valid "low" output voltage (V_{OL_Max}), data frequency is 180MHz and the clock frequency is 200MHz. Assume room temperature operation.

As a validation step, experimentally measure the failure rate and convert it to MTBF.

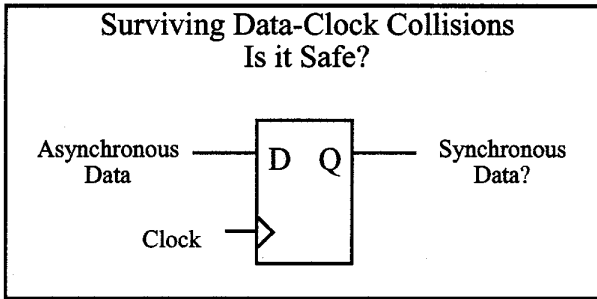


Figure 1. Edge Triggered D Flip-Flop where Data and Clock are asynchronous.

Fortunately, an equation (with a few variations) that relates MTBF (refs [1] and [3]) with these operating conditions has been developed. This is presented in these two ways:

$$MTBF = \frac{t_r}{e^{\tau} 2f_c f_d W} \quad \text{Eq. 1}$$

or

$$MTBF = \frac{10^{\tau_d} t_r}{2f_c f_d W} \quad \text{Eq. 2}$$

where:

- f_c = Frequency of the Clock
- f_d = Frequency of the Data
- W = Metastability Window
- t_r = Resolve Time Available
- τ = Resolving Time Constant
- $\tau_d = \tau/\log(e)$

If you are not planning to determine the Resolving Time Constant, τ , yourself, you should take care to determine for which formula it was derived, equation 1 or 2. Equation 2 is handy as decades (powers of 10) are more convenient when dealing with things like semi-log graphs.

Note that f_d is the frequency of the data as if measured by a frequency meter. It is not the number of data events. The data is changing twice per cycle (e.g. 0 to 1 and 1 to 0) of that measured by a frequency meter. MTBF requires the number of data events. Hence the "2" in the denominator.

In addition to the given conditions of f_d and f_c , device dependent coefficients of W and τ need to be determined. Also, a closer look at the meaning of phrase "...at some time, t , after the clock edge" is needed.

Neither equation 1 nor 2 show the desired time, t . Instead the Resolve Time, t_r , is used. Its definition and accurate determination is important as it is used in the exponent of these expressions (ref [4]). Resolve Time is the time that follows the metastable event. So, when does this event occur?

By definition, it occurs after the Clock edge at a time equal to the maximum allowed propagation delay of Clock to Q Output, $t_{p_cq_max}$. See figure 2.

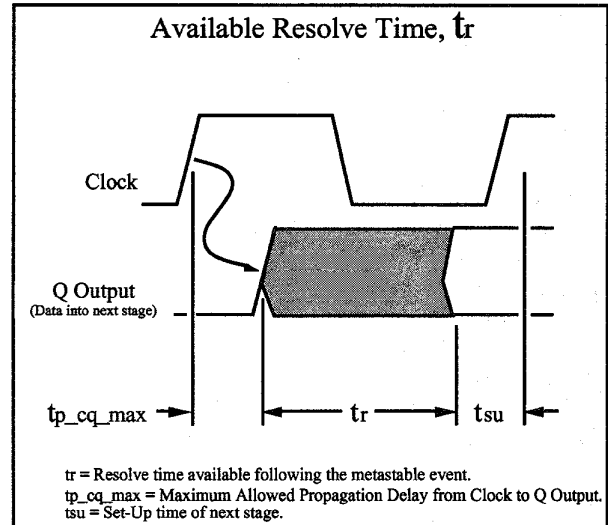


Figure 2. Available Resolve Time, t_r , is the time between the start of the metastable event and the point in time where the Q Output must recover to a legal logic level.

If you ignore the propagation delay to determine Resolve Time, t_r , you will find a significant difference between the predicted and measured MTBF for small values of t_r . That difference will yield a predicted MTBF that is much higher than you will observe.

Assuming that the Q Output will be sampled on the next clock event, Q Output must be resolved prior to the next clock by the Set-up time as given in the manufacturer's specification. For this part, the manufacturer states that $t_{su} = 0.7\text{ns}$ and $t_{p_cq_max} = 1.7\text{ns}$. Therefore, using equation 3, the available resolving time, t_r , is 2.6ns.

$$t_r = (1/f_c - t_{p_cq_max} - t_{su}) \quad \text{Eq. 3}$$

where:

- f_c = Frequency of Clock
- $t_{p_cq_max}$ = Max Clock to Q Output Delay
- t_{su} = Set-up Time of next stage

Finding the Metastability Window, W

The device dependent coefficient W , is found by measuring the Propagation Delay (t_{p_cq}) for different Set-up (t_{su}) and Hold (t_h) times. Figure 3 shows the familiar "static" relationship between these. This is a simple, convenient and safe way to describe a flip-flop's error free sampling relative to the occurrence of the Clock edge. To find the limits where metastable failure occurs, the following characterization can be performed.

The objective of this characterization is to reveal the behavior of the Propagation Delay (t_{p_cq}) as a function of Set-up (t_{su}) and Hold (t_h) times. Figure 4 depicts the anticipated results (ref [1]) where the propagation delay is well behaved (nearly constant) until either the Set-up or Hold times become very small. It is here that the propagation delay begins to increase rapidly. The two points at which the propagation delay exceeds $t_{p_cq_max}$ defines the Metastability Window, W .

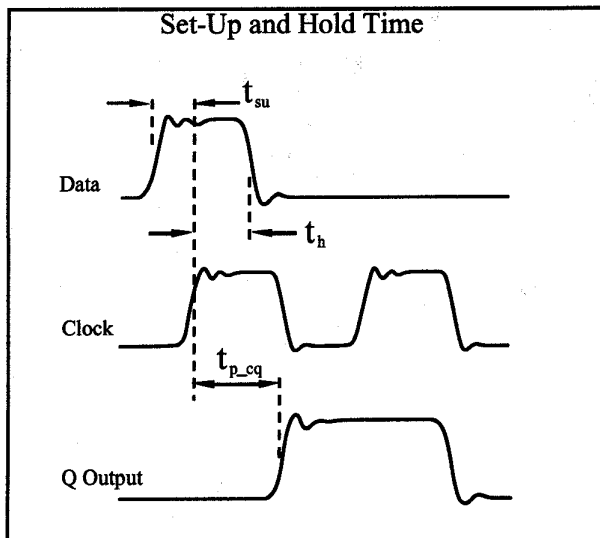


Figure 3. The familiar timing relationship of Data, Clock and Q Output. Accurate measurements of Set-up (t_{su}), Hold (t_h) and Propagation Delay (t_{p_cq}) times are needed.

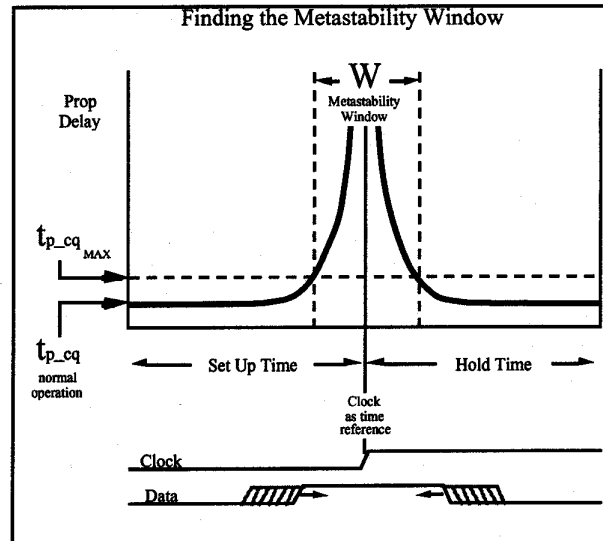


Figure 4. The Metastability Window, W , can be determined by accurately measuring the Propagation Delay time (t_{p_cq}) at different Set-up (t_{su}) Hold (t_h) times. The value for $t_{p_cq_max}$ is given in the data sheet or chosen by the designer as the maximum tolerable Clock to Q Output delay.

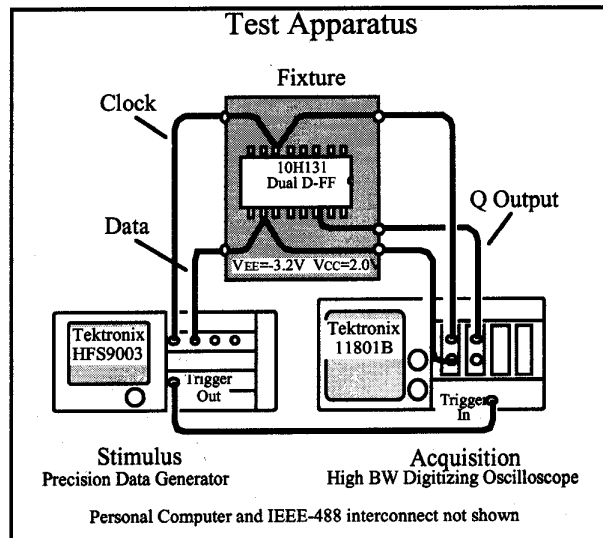


Figure 5. Test apparatus for measuring Set-up (t_{su}), Hold (t_h) and Propagation Delay (t_{p_cq}) times.

Using the apparatus found in figure 5, the set-up time is varied relative to the clock as in figure 6, the set-up and propagation delay times at the flip-flop are measured and plotted. Similarly, hold time is varied as in figure 7 and the hold and propagation delay times are measured and plotted. As the prop delay begins to increase, smaller increments in the edge position of the Data signal are used. Initially, 100ps increments are used. By the time the Data is positioned well within the Metastability Window, increments as small as 5ps are used. Locating the window to within a few tens of picoseconds is adequate for this device.

The Clock and Data signal pattern produced by the HFS9000 is actually a burst of one Data pulse with two Clock pulses as shown in figures 6 and 7. The burst repetition rate is between 5ms and 10ms. The interval of the two Clock pulses needs to be long enough for the metastable to be nearly "fully" resolved. Otherwise the second Clock pulse of the burst will reset the Q Output too soon which will interfere with your observation of the metastable decay. For this experiment, the interval was 20ns.

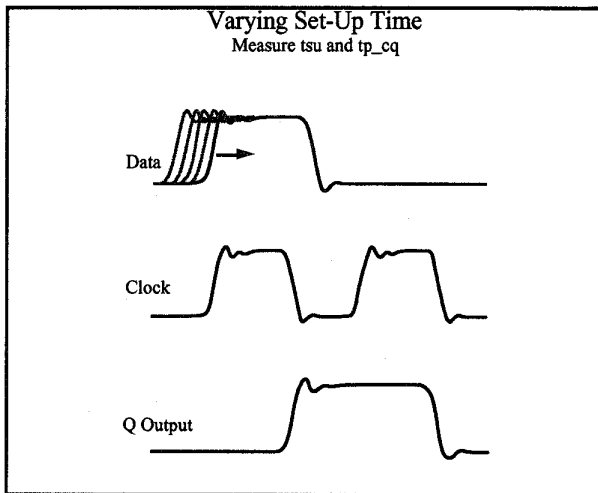


Figure 6. To determine the behavior of Propagation Delay (t_{p_cq}) as a function of Set-up (t_{su}), the precision timing generator in figure 5 positions the edge of the Data signal toward the Clock signal.

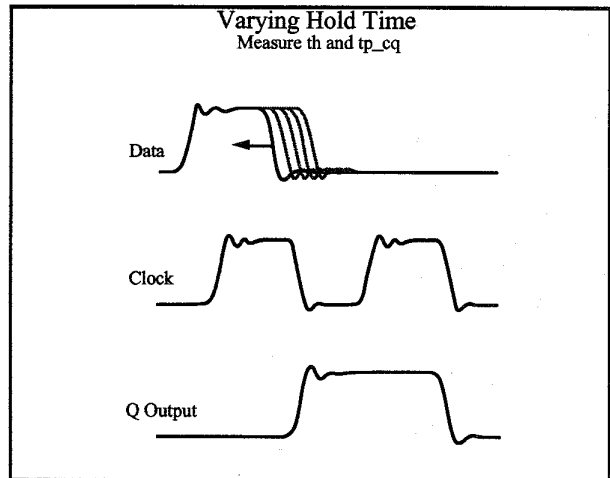


Figure 7. Similar to figure 6, to determine the behavior of Propagation Delay (t_{p_cq}) as a function of Hold (t_h), the precision timing generator in figure 5 positions the edge of the Data signal toward to the Clock signal.

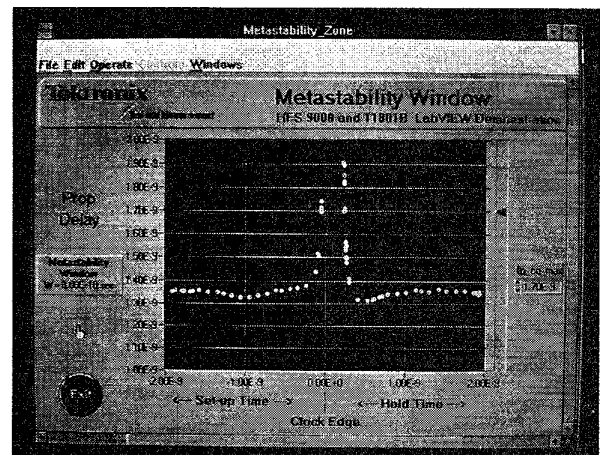


Figure 8. A plot of Propagation Delay (t_{p_cq}) as a function of Set-up (t_{su}) and Hold (t_h) times.

Figure 8 is an actual plot of Propagation Delay vs Set and Hold time with the intercept indicator at $t_{p_cq_max}$. Table 1 shows the value of W, Metastability Window, for the four different 10H131 flip-flops.

10H131P D-FF	
Lot Date	W
KKDI 8906	170ps
FFRQ 9111	220ps
KKHQ 9137	225ps
KKEF 9140	206ps

Table 1. Metastability Window.

Finding the Resolving Time Constant (τ)

The Resolving Time Constant, τ , that is found in the MTBF equations 1 and 2 comes from the expression that describes the probability of a metastable event lasting longer than some time, t (ref [3]). This probability is expressed as:

$$P = e^{-t/\tau} \quad \text{Eq. 4}$$

or, if you prefer the decimal form

$$P = 10^{-t/\tau_d} \quad \text{Eq. 5}$$

To experimentally determine the Resolving Time Constant, τ , metastable events need to occur at a fairly high rate so that the data can be gathered in a reasonable amount of time (a few minutes). Rather than the traditional approach of using two unsynchronized signal sources for Data and Clock signals, a stable synchronous source can be used. Again, with the same apparatus shown in figure 5, the placement of the Data edge can be precisely controlled such that nearly every Clock produces a metastable event. As both Data and Clock signals are synchronized within the HFS9000, it is possible to place the Data edge in the center of the Metastability Window. This is accomplished by observing the Q Output while adjusting the placement of the Data relative to the Clock until the Q Output appears to favor both high and low logic levels equally. See figure 11.

Now you need some way to observe and analyze the "decay" of this metastable behavior over time. For this the 11801B is used for its Infinite Persistence display and Mask testing. The concept of Mask testing is shown in figure 9. Masks are zones (up to 50 sided polygons) that are defined and placed in different locations on the screen. If a waveform sample falls within a mask, a counter is incremented. The tally for each mask counter is displayed at the bottom of the screen. In this way, the samples falling in one mask can be compared with the samples falling in another mask. In figure 9, two masks are defined that are of the same vertical and horizontal size. This means that they both represent the same span of time and voltage. The top and bottom of each mask is set to a level equal to the valid "high" output voltage (V_{OH}) and the valid "low" output voltage (V_{OL}) respectively. When placed at different time locations overlaying the Q Output of a flip-flop forced into metastable operation, the two mask counts reveal that there are fewer unresolved occurrences later in time than earlier in time.

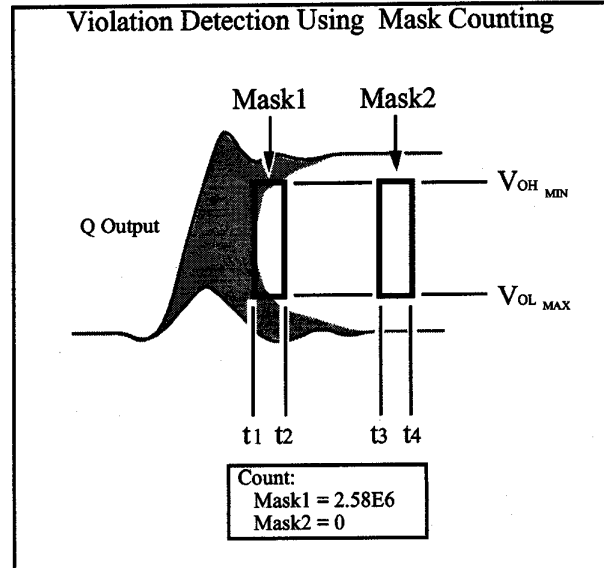


Figure 9. User defined zones are can be created to accumulate logic state violations at different times following the Clock.

As there are ten masks available, they can be distributed at different times to tally the illegal samples as a function of time. The construction of these masks are a little different than in the example of figure 9. Because you want to observe the population at some time, t , and later, the masks must start (left-hand side) at later times but all end (right-hand side) at the same time. See figure 10.

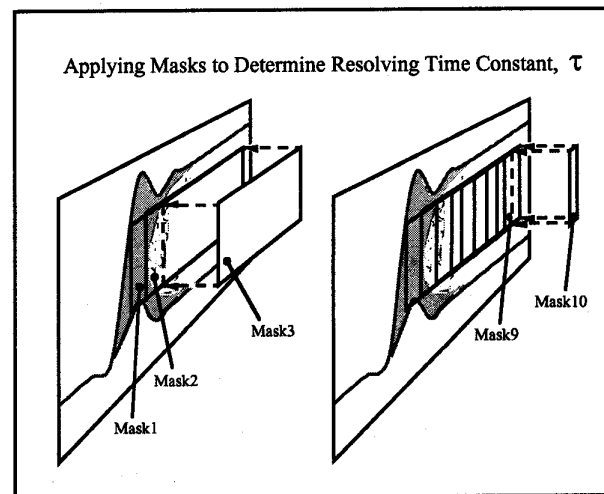


Figure 10. Mask Counting can be applied to determine the population of violations as a function of time.

The tallies of these masks will reveal the population decay rate as expressed in equations 4 and 5. Figure 11 shows the 11801B displaying the masks overlaying the Q Output of the flip-flop.

The starting location of each mask should be offset either 0.5 divisions or 1.0 divisions. Smaller than 0.5 and you may not be using the samples efficiently. That is to say that a lot of samples are wasted as they will not help to discriminate the time related behavior. It may also take longer to acquire a sufficient sample size. Larger than 1.0 division and you can't get ten Masks on the screen!

The top and bottom of the masks should correspond to the legal logic levels for your logic family, such as the minimum valid "high" output voltage (V_{OH_Min}) and the maximum valid "low" output voltage (V_{OL_Max}).

The time/division setting of the 11801B should be chosen such that there is a substantial decay captured within the masks. See figure 11. The Q Output should be nearly or completely resolved to legal logic levels at the screen locations corresponding to Mask10 (far right-hand side of the display).

The Q Output waveform should be positioned as shown in figure 11. The beginning of the metastable event should be near to the left-most mask edge. Although it is important to position the masks early in time, near to the metastable event, it is not critical to start exactly at the metastable event. It is important that the masks are positioned after the start of the event and not before. Remember that all that is determined by this method is the slope of the decay and not the absolute total error at some time. A method for that is discussed later.

Finally, the masks must have vertical sides such that each mask clearly defines an unambiguous time interval. Slanted sides will cause samples to be incorrectly associated with a time interval depending upon their vertical location.

Of course, the masks can be created and their tallies, as shown in figure 12, can be retrieved through the 11801B's IEEE-488 interface.

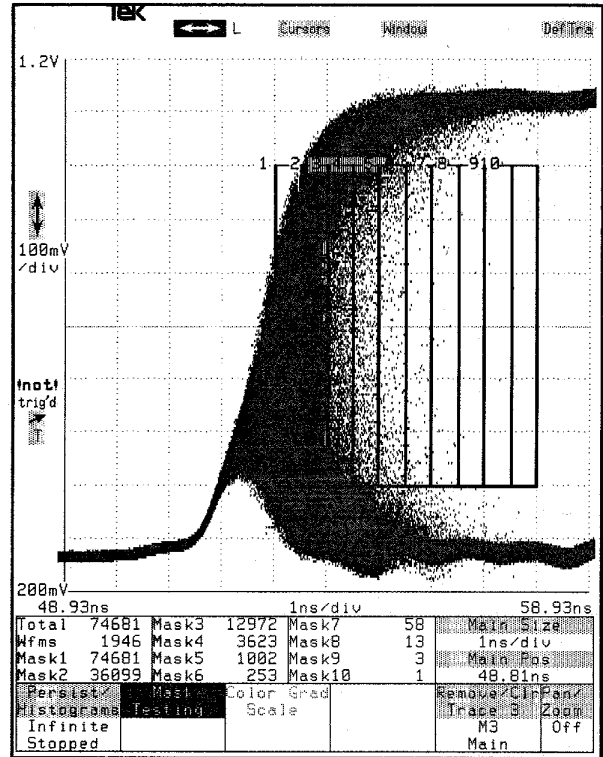


Figure 11. An 11801B with masks constructed and overlaid on the Infinite Persistence display of the Q Output.

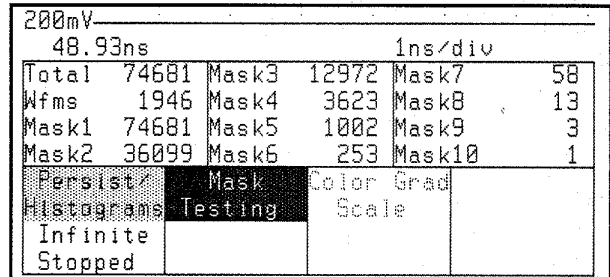


Figure 12. After a few minutes of collecting samples from the metastable condition, the mask counts have acquired sufficient information to determine the Resolving Time Constant, τ .

The data are normalized by computing the ratio of each mask count to the count of Mask1. Mask1 represents the entire population of illegal samples. In a semi-log plot, the population decay ($1/\text{Resolving Time Constant}$, $1/\tau_d$) is revealed as the slope of the line. See figure 13.

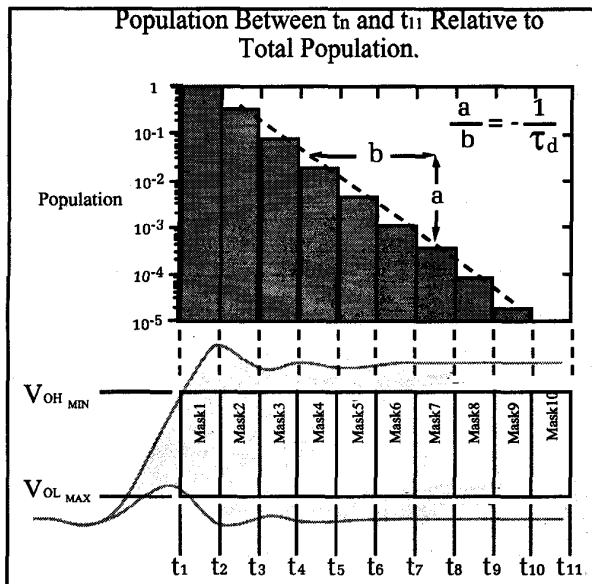


Figure 13. The mask counts are normalized to the count of Mask1(entire violation population). The log of these ratios are plotted versus time. The slope, a/b , is equal to $-1/\tau_d$.

The result from actual data is shown in figure 14. Table 2 lists the value for Resolving Time Constant, τ , for the four different 10H131 flip-flops.

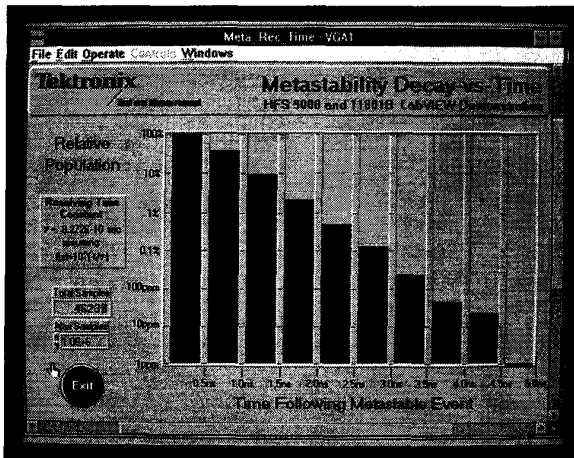


Figure 14. A plot of the Metastability Decay versus Time.

10H131P D-FF		
Lot Date	τ	τ_d
KKDI 8906	393ps	906ps
FFRQ 9111	399ps	918ps
KKHQ 9137	393ps	905ps
KKEF 9140	369ps	849ps

Table 2. Resolving Time Constant.

Predicting the MTBF

Assuming that the device dependent coefficients (W and τ) were carefully gathered, you are ready to compute the MTBF. Using equation 2, a value of $\tau_r = 2.6\text{ns}$ and the data from tables 1 and 2 yield the results in table 3.

10H131P D-FF				
Lot Date	τ_r	τ_d	W	MTBF
KKDI 8906	2.6ns	906ps	170ps	61 μs
FFRQ 9111	2.6ns	918ps	220ps	43 μs
KKHQ 9137	2.6ns	905ps	225ps	46 μs
KKEF 9140	2.6ns	849ps	206ps	78 μs

Table 3. Predicted MTBF using manufacturer's values for maximum prop-delay and set-up times.

Measuring Failure Rate and Computing MTBF

To experimentally measure the average failure rate and thereby compute MTBF, the test apparatus was changed by substituting a Data source that is asynchronous to the Clock. See figure 15.

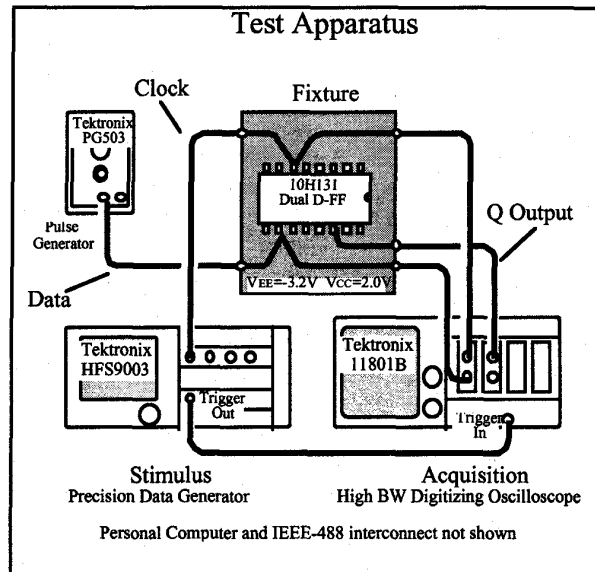


Figure 15. Test Apparatus for directly measuring Failure Rate of asynchronous operation of the flip-flop. A high rep-rate pulse generator was added to the original configuration of figure 5 to serve as the asynchronous Data source.

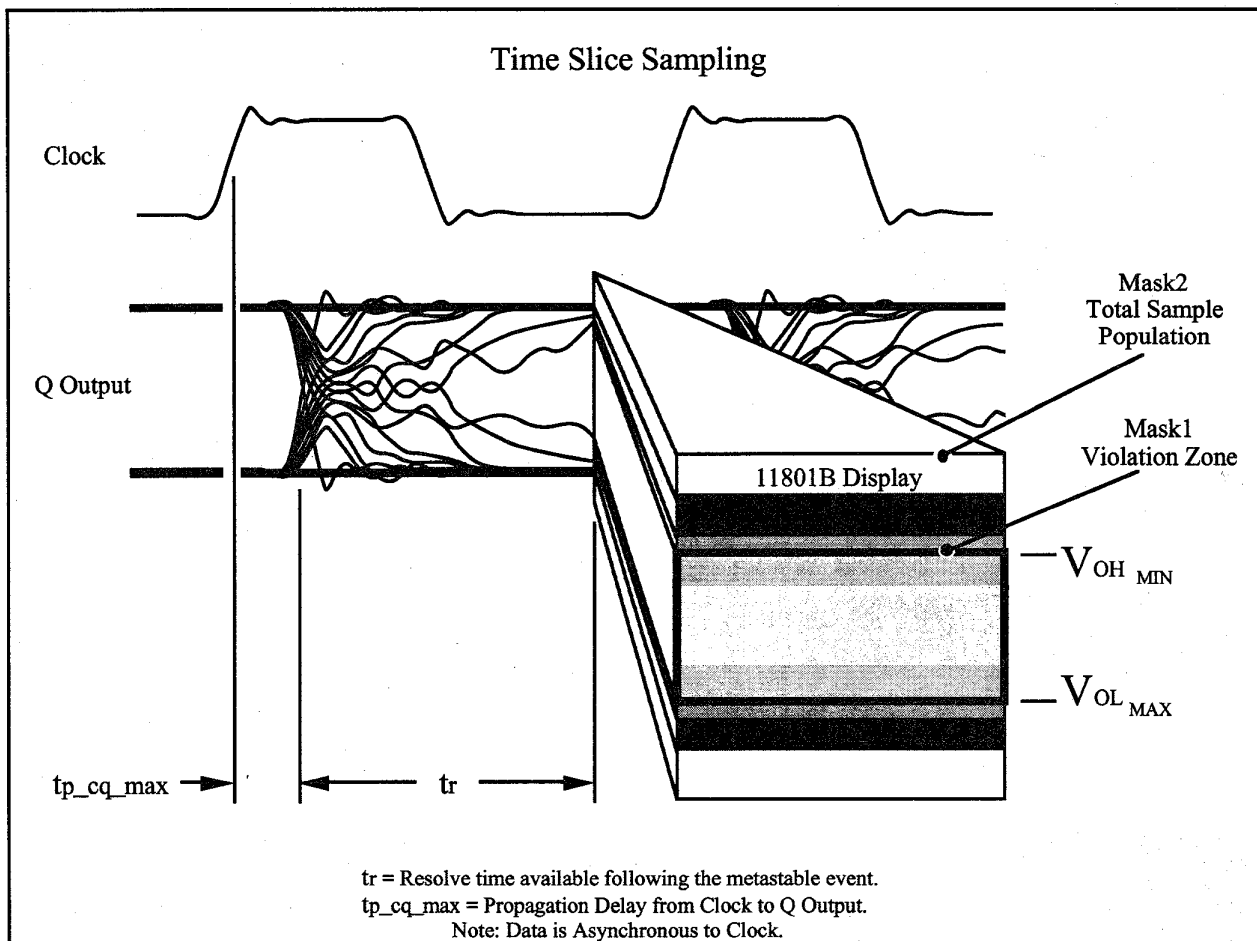


Figure 16. Observing the failures at a point in time is possible with the 11801B. Its fastest time/div is zero! That means that the entire horizontal span of the display depicts samples taken at the same point in time. This is, in effect, an extremely narrow slice in time. The position of this time slice is controlled by the Main Position setting which controls a precision, ultra-low jitter delay generator referenced to the Trigger input. Masks count violations (illegal logic levels) relative to the total samples taken (entire screen).

To observe what occurs at the time, t , after the Clock, the 11801B offers a unique ability to choose that point in time and acquire waveform samples only at that time. This is accomplished by selecting 0s/div as the time scale factor. Yes indeed, that is zero time span. This means points that appear anywhere on the display were all taken at the same point in time. This constitutes an infinitesimally narrow slice of time plus or minus a few picoseconds due to jitter. In the case of the 11801B, at 0s/div, the delay increment that normally advances the sample point in time has been disabled.

Sampling at this "slice-in-time" allows you to apply Mask testing once again to tally violations and compute a failure rate at this point in time. See figure 16.

For this experiment, only two masks are needed. Mask1 is the violation zone and Mask2 overlays the entire population of samples. Therefore the tally of Mask1 divided by the tally of Mask2 is the average failure rate. See figures 17 and 18. The failure rate is very sensitive to the placement of Mask1's top and bottom levels. So care must be taken when choosing your definition of a failure. It is also critical that you prevent samples from exceeding the vertical range of the display by choosing a vertical scale factor that covers the dynamic range of the flip-flop's Q Output. Points that fall above or below the vertical range of the display are discarded. These might otherwise be counted as good values. By omitting these from the total population, you will bias the results in favor of a higher failure rate (lower MTBF).

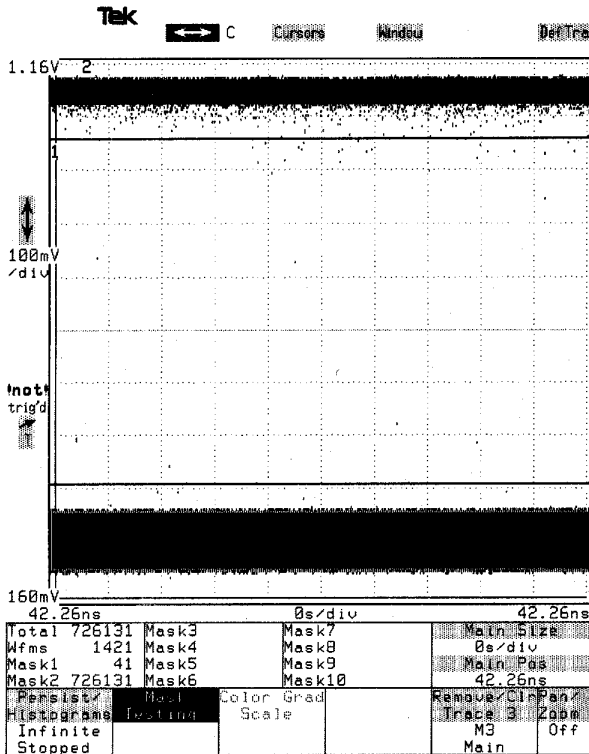


Figure 17. The 11801B set at 0s/div to observe a point in time after the Clock as shown by the Main Pos. Two masks are used. Mask1 defines the violation zone. Mask2 covers the entire acquisition range as it gathers the total samples counted at this point in time.

Total	726131	Mask3	Mask7
Wfms	1421	Mask4	Mask8
Mask1	41	Mask5	Mask9
Mask2	726131	Mask6	Mask10

Figure 18. The ratio of Mask1(violations) to Mask2(total samples) yields the rate at which the Q Output fails to resolve to a valid logic level.

10H131P D-FF					
Lot Date	Total Samples	Violations	Observed MTBF	Predicted MTBF ($t_r = 2.6ns$)	Predicted MTBF ($t_r = 3.0ns$)
KKDI 8906	4.41E6	100	220 μs	61 μs	167 μs
FFRQ 9111	3.12E6	100	156 μs	43 μs	117 μs
KKHQ 9137	1.93E6	100	96 μs	46 μs	127 μs
KKEF 9140	2.78E6	100	139 μs	78 μs	230 μs

Table 5. Comparison of Measured and Predicted MTBF

Table 5 shows the result of the failure rate measurements and compares them to the predicted MTBF.

$$MTBF = \frac{\text{Total Samples}}{(\text{Violations}) f_c} \quad \text{Eq. 6}$$

where:

- f_c = Frequency of the Clock
- Total Samples = Mask2 Count
- Violations = Mask1 Count

You will notice that the observed and predicted MTBF ($t_r = 2.6ns$) differ by as much as four times. The exponential nature of the MTBF formula is extremely sensitive to small changes in t_r and τ . The value for t_r was computed using the manufacturer's specification for the maximum prop. delay, $t_{p_cq_max}$. In this experiment a pronounced metastable condition occurred much earlier than the stated maximum. The actual metastable condition began near the typical prop. delay. This was about 0.4ns earlier than the specifications suggested. This has the result of increasing the value of t_r to 3.0ns. Table 5 shows two predicted MTBF values to show the impact of the different t_r values. The MTBF using $t_r = 3.0ns$, agrees within a factor of less than two. Also, the predicted MTBF is an estimate of the statistical mean and assumes that the data and clock are truly random. In designing a reliable synchronizer, it is hoped that the resulting performance is measured in orders of magnitude in improvements. This technique, without further refinement, is certainly suitable for gathering reliable values for use in predicting the performance of a synchronizer.

Conclusion

Some of the techniques presented here are analogous to Peter Stoll's (ref [1]) popular "Late Transition Detector." Masks can be positioned both vertically and horizontally which is the same as setting voltage thresholds and adjusting time delays. The mask tallies are like gated event counters. When coupled with the advances in precision instrumentation over the last ten years, these analogous features offer better control and greater accuracy in defining and trapping violations and can be applied to a wider variety of logic families. If you need to test ECL after testing CMOS, you won't have to create, debug, calibrate and validate yet another custom violation/MTBF detector. Creating copies of that custom hardware that have the same performance may pose additional problems that delay you from gathering reliable information and improving the quality of your designs.

References

[1] Stoll, Peter A., "How to Avoid Synchronization Problems," VLSI Design, Nov/Dec 1982, pp 56-59.

[2] Bolton, Martin, "A Guided Tour of 35 Years of Metastability Research," Professional Program Session Record 16, WESCON 87, Nov 1987, pp 16/4, Electronic Conventions Management, Los Angeles, CA 90045.

[3] Dingman, Sean, "Are Your PLDs Metastable?" Application Note, 17 July 1992, Cypress Semiconductor Corp., 3901 North First Street, CA 95134.

[4] Nootbaar, Keith, "Design, Testing, and Application of a Metastable-Hardened Flip-Flop," Professional Program Session Record 16, WESCON 87, Nov 1987, pp 16/2, Electronic Conventions Management, Los Angeles, CA 90045.