Measuring the Distribution of Metastable Upsets over Time

Thomas Polzer and Andreas Steininger
TU Wien, Institute of Computer Engineering
A-1040 Vienna, Austria
{*thomas.polzer, andreas.steininger*}*@tuwien.ac.at*

***Abstract*—As modern ASICs comprise an increasing number of independently clocked subsystems that need to interact, the accurate reliability assessment of synchronizers becomes crucial. Traditionally the reliability of a synchronizer is characterized by the mean time between upsets (MTBU), and the relevant flip-flop parameters are specified in a way to support MTBU calculation.**

In this paper we claim that actually a deeper insight into the distribution of upsets over time is needed in order to make a reasonable prediction in the range of the high reliability values that are generally targeted. We present a measurement concept that appropriately extends state-of-the-art approaches so as to allow for an experimental assessment of the upset distribution over time. In this way the distribution function can be studied, and in particular the probability of upsets with low temporal distance – which is the relevant one for high reliability – can be identified. We implement our concept on three different FPGA platforms and present the selected results. The distribution function we obtain indicates that the assumption of a uniform or standard normal distribution, which one might be tempted to imply for lack of better information, is definitely not generally useful.

***Keywords*-late transition detection, metastability measurement, FPGA, time between upset, distribution**

I. Introduction

For today's ASICs comprising billions of transistors that operate at GHz clock rates, the globally synchronous design paradigm cannot be sustained anymore. Instead GALS (globally asynchronous locally synchronous) architectures [1] have become the de facto standard. Here the architecture is composed of several function blocks which are independently clocked. Communication between these blocks is typically based on some kind of asynchronous handshaking principles. However, these handshake signals necessarily cross clock domain boundaries and hence need to be synchronized. As a consequence, synchronizers become dominant building blocks in modern systems-on-chip, and their optimal design and modeling becomes crucial. After all, there is an inherent trade-off between the reliability of a synchronizer and the performance penalty it incurs. In this context the characterization is traditionally based on the calculation of the synchronizer's mean time between upsets (MTBU), which is the expected time until a first synchronization failure occurs after putting the system in operation (or the next failure after a previous one). This is actually not the question one usually has to answer in the reliability analysis of a system. There, rather than knowing that the failure will occur on average every 1000 years or so, it would be much more interesting to calculate the probability of such a failure to occur within the life time (or mission time) of the system. Of course, this probability can be derived from the MTBU, but only if the underlying distribution is known. In spite of this circumstance, the existing approaches determine the MTBU (or the relevant flip flop parameters, namely the metastability resolution constant τ and the aperture window T_0) more or less by its direct measurement, without giving any insight into the actual distribution of failures over time.

In this paper we will present a novel measurement approach, along with a proof-of-concept implementation, that allows to determine the actual distribution of synchronization failures over time. We will present selected experimental results that we obtained for three different FPGAs and that show that this distribution is definitely not a uniform or standard normal distribution which one might be tempted to imply when being supplied with an MTBU only. The paper will be structured as follows: In the next section we will give a background on metastability and MTBU estimation, as well as present the state of the art measurement approaches. Next, we will present our new concept in Section III, and illustrate its implementation in Section IV. Section V will then be devoted to experimental measurements on an FPGA that demonstrate the applicability of our approach and give a first insight into the actual distribution of synchronization failures over time. Finally, Section VI will conclude the paper and give an outlook on future work.

II. State of the Art

A. Metastability and MTBU

Metastability occurs when a bistable element gets caught in an equilibrium between HI and LO output state. For a synchronizer flip flop this situation may occur when data and active clock edge are not sufficiently separated, as specified in the setup- and hold time. When moving a signal across the boundaries of two independent clock domains, such a situation cannot generally be avoided [2], [3]. For the flip flop to leave the metastable state, it takes an extended (essentially unbounded [4] but usually small) amount of time, during which the output is undecided. Should this undecided output state cause an incorrect or undecided value to be captured by a subsequent bistable element, we call this a metastable upset. For uncorrelated clocks, and in the presence of noise and jitter, the occurrence of such upsets is indeterministic and therefore often considered "random". However, it is reasonable to assume that it follows some statistical distribution, and the common way of characterizing this distribution is by way of the mean time to upset (MTBU). The widely used equation in this context is [5], [6]

$$MTBU = \frac{1}{\lambda_{dat} f_{clk} T_0} \exp\left(\frac{t_{res}}{\tau}\right) \tag{1}$$

where λ_{dat} is the rate of transitions at the data input, f_{clk} the function block's clock frequency, t_{res} the time allowed for the metastable state to resolve, and T_0 and τ the flip flop's characteristic parameters (technology dependent). It is fundamentally based on the assumption of uniformly distributed phase of data transitions relative to active clock edges of the synchronizer flip flop.

It is the purpose of a synchronizer to keep the MTBU sufficiently high, which is typically achieved by allowing a long resolution time t_{res}. With multistage synchronizers an arbitrarily high MTBU can be achieved, just by adding further stages. However, each stage introduces a performance penalty, so a designer is interested in keeping the number of stages at a minimum. When it comes to choosing an appropriate MTBU borderline, one is usually very cautious by selecting a value of, say, 1000 years or even more, and then more or less neglects the probability of a metastable upset in the further analysis. However, this is problematic for two reasons:

- As we are interested in optimizing the trade-off between reliability and performance we do not want to be overly conservative. So it would be good to know which MTBU target value is actually required.
- Even if we deem ourselves conservative – what does an MTBU value of, e.g., 1000 years actually mean for our reliability? In systems with high reliability demands it may well project to an insufficient reliability value, if the temporal distribution of upsets has a very high variance and hence the probability of a short time between upsets becomes sufficiently high. Without knowing the actually underlying distribution we cannot reasonably form a relation between MTBU and reliability.

As a consequence we claim that it is necessary to investigate this distribution. It may well turn out that, once it is sufficiently understood, we may continue relying on MTBU parameters only; but to attain this understanding we need measurement circuits that provide more detailed results. The contribution of this paper is to present a circuit that is suitable for this very purpose.

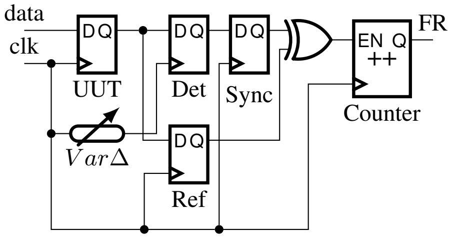


Figure 1: Basic LTD circuit

B. Measurement of metastability parameters

The traditional [7], [8] circuit for determining the metastability behavior of a flip flop is shown in Figure 1. Herein the flip flop under investigation (UUT) is provided with uncorrelated clock and data such that it occasionally becomes metastable. The clock frequency is chosen low enough to allow for sufficient resolution time within one full clock period in practically all cases. Consequently, the reference flip flop (Ref) captures the UUT's correct output value after metastability has properly resolved. This reference value is compared to the output of the detector flip flop (Det). Since the latter is provided with a delayed clock signal, it samples the output of the UUT after that delay (VarΔ) already, thus allowing for very limited resolution time. By comparing the reference output with the detector output it can be determined whether the UUT's output had already resolved when sampled by Det, i.e. after VarΔ. This decision is made by the XOR gate (after synchronizing Det's output to the reference through Sync), and all mismatches are counted as upsets. The upset rates are determined for different resolution times, and by plotting the resulting function in a semi-logarithmic scale, the metastability resolution time constant can be derived from the linear slope.

This circuit principle has been extended in [9] to allow for a more detailed analysis. In particular, the (usually different) metastability resolution constants of master and slave latch [10] can be identified. This is achieved by performing multiple measurement runs with different HI duration of the clock pulses, which causes the slave to become metastable at different values of t_{res} and thus allows a clear distinction in the diagram. Additionally [9] proposes a strategy for separating the following different metastability manifestations: (1) late up-transition, (2) late down-transition, (3) positive glitch, and (4) negative glitch. The key for this is to remember the previous state of the UUT in a flip flop (Last in Figure 2) and, based on a comparison between current and previous state direct the observation of an upset to a counter that is specific for the type of event. Figure 2 shows the related circuit. It can be seen that the counter in the original circuit has been replaced by a logic for the case distinction, and that now five counters are present, one for each of the cases plus one for the overall

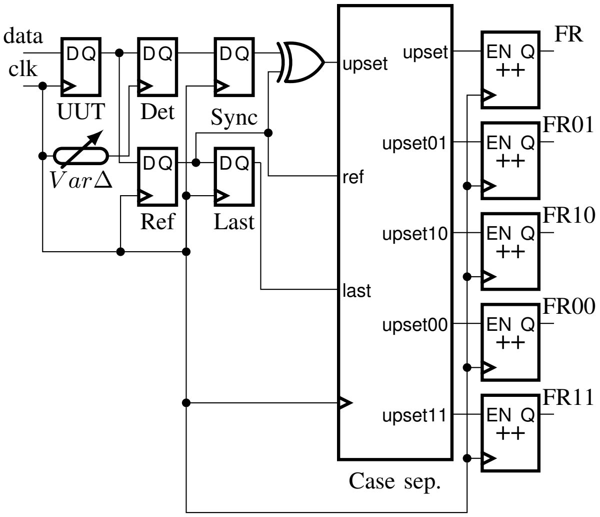


Figure 2: LTD circuit with case separation

sum[1].

Note that, while allowing a much more detailed analysis, this circuit still aggregates all observations of upsets in counters. This strategy is sufficient for calculating the required upset rates and, consecutively, the MTBU, but it discards all the concrete information about the relative distance between subsequent upsets. The latter information is, however, exactly what we need for our purpose, therefore we will present a respective improvement in the following.

III. Propoesd Measurement Concept

We base our measurement concept on the work presented in [9] and extend the circuit accordingly. In particular, our measurement circuit uses the same late transition detection scheme and case distinction, and it is also able to modulate the positive clock pulse length to generate different master-slave ratios. The key difference is that in our proposed circuit we replace the counters by some more elaborate function blocks that in addition to the number of observed events also provide their time of occurrence. To this end we need to establish a time base, and we need to provide a means to produce a time stamp whenever an upset is observed. These time stamps must be stored and transferred to a host computer. Figure 3 illustrates how we extend the measurement concept from [9].

The time base is implemented as a counter that is increased by one with each reference clock tick. The counter output is connected to multiple FIFO memories, one for each metastability case. If the case separation circuit indicates the presence of an upset of a particular type, the corresponding FIFO memory is enabled for one clock cycle and stores the corresponding output of the time base counter. The width of the counter must be large enough to ensure that we do not get ambiguous time stamps due to its wrap-around. More specifically, if we store a time stamp T_x then the counter must not reach the same value T_x' again after wrap around, without having issued another time stamp to the same FIFO in the meantime.

Basically the capturing of time stamps through the measurement circuit and their read-out through the host are done in parallel, and the FIFO is only there to compensate for temporary speed differences. However, in phases with a high upset rate the resulting high rate of time stamps may exceed the bandwidth of the host interface. In that case the FIFO becomes full sooner or later (depending on its depth), and in order to avoid having “blind spots” with seemingly no upsets in the sequence of time stamps, the recording must be completely disabled and no later time stamps be accepted. Note, however, that this can be done on a per-FIFO basis, therefore FIFOs that see a smaller rate of time stamps may remain in operation longer. This is important to still identify relatively rare upset types within a stream of dominating other types.

The purpose of the communication interface is to transfer the streams of time stamps from the FIFOs to the host PC. This involves a higher bandwidth than in the original circuit where only the counter values had to be read once the measurement was over. For a standalone solution it would, in principle, be possible to implement the required data processing directly on chip, using a state machine approach or a soft core processor. This would reduce the required bandwidth to a couple of single values per measurement run.

The measurement procedure is at follows:

1) First the detector is calibrated to find the UUT’s nominal output delay by using a synchronous data signal. The detector delay varΔ is tuned to a point where it detects no upset but, if the delay is shorted by one step, upsets will occur.
2) A predefined number of steps is subtracted from the calibration value to ensure that the measurement will start within the nominal output delay and the transition to metastability is recorded.
3) The measurement is started with that delay value, and the timestamps of all upsets are recorded. The measurement is stopped when both the recorded sums of rising late transition and positive pulses as well as the sum of falling late transition and negative pulses exceed 200. If one or more of the FIFOs overrun in this process, the measurement of the overrunning FIFO is stopped at that time, but all other measurements are continued.
4) After a measurement step has finished, the detector delay varΔ is increased by one step and the measurement for the new step is executed according to step (3).

[1] Note that while the case separation gives a good insight for analysis purposes, the MTBU is still determined by the overall sum of effects.

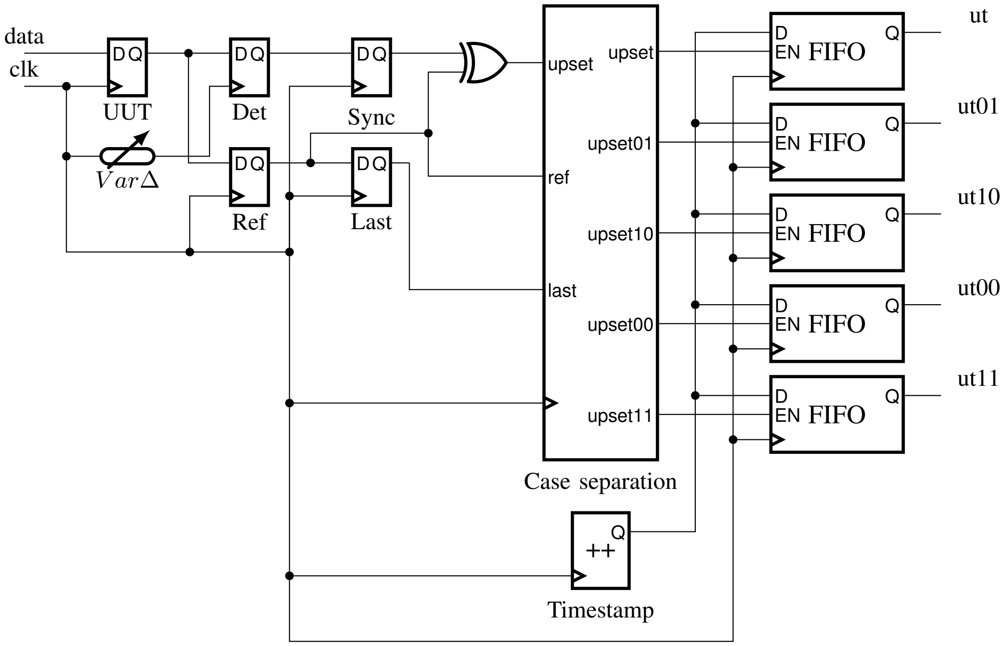


Figure 3: LTD circuit with time stamp recording

Note that the calibration in steps (1) and (2) are the same as proposed in [9] (for details see there), while the measurement step (3) has been adapted to our specific purpose.

IV. Implementation

We have implemented the proposed scheme on three different target platforms, namely a Xilinx Virtex4 90nm FPGA, an Altera Cyclone IV 60nm FPGA, and a Xilinx Kintex-7 28nm FPGA. The basic circuit for all three implementation is the same:

- For monitoring and controlling the measurement a python script was implemented running on a standard PC. The target device was connected to the monitoring PC using a standard RS232 serial port.
- For the time base counter implementation we chose a width of 48-bit. At 100MHz this yields a wrap around every 782 hours, which is more than sufficient but still reasonably cheap to implement.
- The FIFOs have a storage capacity of 2048 time stamps.

The main difference between the three platforms lies in the implementation of the clock generation circuit and the variable delay line. They had to be customized to the target device, as specialized components, such as phase locked loops (PLLs) or discrete clock manages (DCMs), are required.

The implementation of the Virtex4 circuit is directly derived from [9]. We use a DCM to multiply the base clock of 100MHz to generate the reference clock of 200MHz. Using an additional DCM we are able to phase shift the detection clock signal in steps of 19.53ps. This resolution is worse than the 9.77ps from the original circuits, because the latter has a less complex evaluation circuit and can hence operate at 400MHz. Unfortunately the step size directly depends on the output clock frequency in this DCM. The clock pulse length generation circuit requires another DCM and an AND mask to shorten the high time of the UUT clock signal.

For the Altera implementation a PLL is used to generate all clock signals. They are derived from a base clock with 50MHz. The three different clock signals (UUT, Ref and Det) are derived with a frequency of 150MHz. Using the fine phase shift functionality of the PLL, the detection clock can be shifted in roughly 104ps steps, while the dynamic reconfiguration mechanism is used to reprogram the clock pulse length of the UUT clock according to the current measurement specifications.

The implementation for the Kintex-7 FPGA is similar. A PLL is used to generate the UUT and reference clock signals with 200MHz. Again a reconfiguration controller is used to adjust the clock pulse length of the UUT clock according to the measurement requirements. As the stepping of the PLL is not as fine as the one of the integrated clock managers, a mixed-mode clock manager (MMCM) was used to generate the detection clock (also 200MHz) signal. We were able to achieve a step size of only 12.56ps in this case.

To achieve optimum results, the timings of critical signals (signals between UUT, Ref and Det) are controlled by virtue of timing constraints. This ensures that the delay, and therefore the undesired reduction in resolution time of the reference flip flop, can be kept at a minimum. It is, however, important to note that too tight timing constraints may cause the implementation tools to fail in ensuring hold times in the fast process corners. In such a case loosening up the timing constraint a bit should correct the problem.

The usage of the PLLs instead of DCMs for generating the UUT clock makes the timing closure much easier to achieve compared to the version using an AND mask (as described in [9]). To be comparable to the original publication, we nevertheless implemented the version for the Virtex4 FPGA in the same fashion as in the original paper. Therefore, in this variant, the signal routing must be controlled much more rigorously than in the other implementations (more placement constraints are required).

V. Results

We executed the measurements on all three targets described in Section IV and received similar results for all of them. For brevity, and since it is the one with the smallest feature size, only the results for the Kintex-7 platform are shown here.

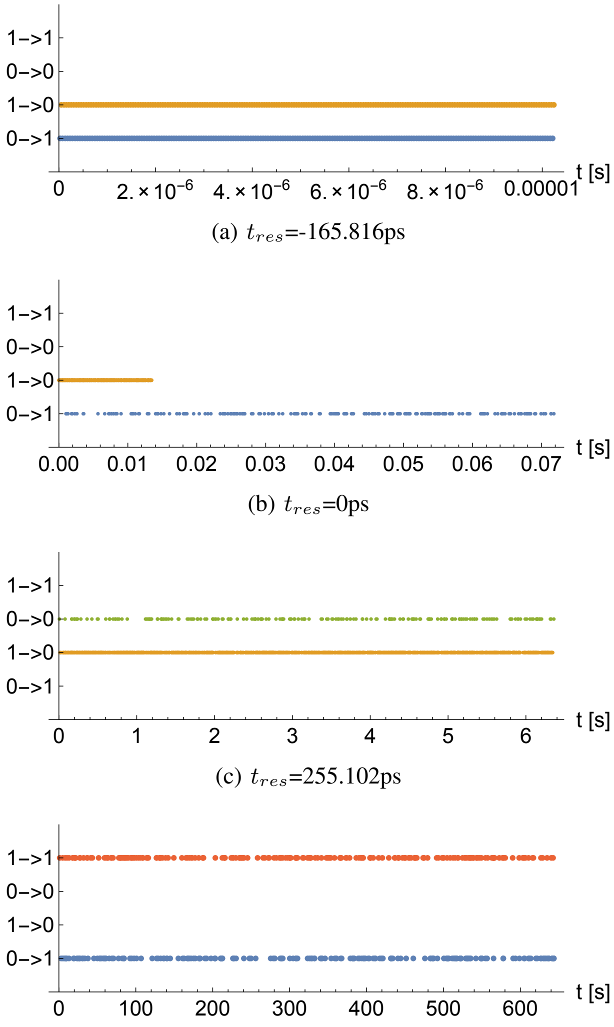
Figure 4 shows selected examples of the recorded data. We see four sub-graphs, each for a different resolution time (indicated in the caption of the respective sub-graph) and with a different time scale (see labeling of the x axis). In these graphs each dot represents one upset with color of the dot and y-position indicating the type of upset, (see labeling of y axis; the case "1 $\rightarrow$ 1" represents a transition starting with 1 and ending with 1, hence a negative glitch, accordingly "0 $\rightarrow$ 0" indicates a positive glitch). In the topmost sub-graph (t_{res}=165.816ps) we see a case with both, late up- and late down-transitions. In spite of the short time interval ($10\mu s$) we observe dense sequences, which points to a high rate of upsets.

The second sub-graph is for zero resolution time, but over a longer observation interval (70ms). Here we see the late up-transitions spread over the whole interval, while the late down-transitions seem to be concentrated to the leftmost area. This indicates that we had a FIFO overflow for the latter, while the FIFO for the up-transitions kept recording larger time stamps. The conclusion here is that the late down-transitions are dominating.

In sub-graph (c) we see a combination of positive glitches and (very frequent) late down-transitions, this time spread over a recording interval of more than 6 seconds. Finally, the sub-graph at the bottom shows an extreme case of nearly 700 seconds observation time, this time with a combination of negative glitches and late up-transitions.

From these graphs we can already draw the following conclusions:

- Our approach works well for very frequent as well as infrequent upset types; the observation time is adapted accordingly.
- In case of very divergent upset rates we get different intervals recorded (due to the FIFO locking).
- The different types of upset can be nicely distinguished and their frequencies be compared.
- The occurrence times of upsets are quite randomly distributed and no clear beat frequency between data and clock signal can be identified. We take this as a confirmation that in our setup clock and data signals are not correlated.



(a) t_{res}=-165.816ps

(b) t_{res}=0ps

(c) t_{res}=255.102ps

(d) t_{res}=688.776ps

Figure 4: Selected measurement results for Kintex-7

The graphs in Figure 4 show the situation for one specific resolution time each. To be able to better analyze and understand the metastability resolution process and to compress the recorded data into a more comprehensible form, we decided to aggregate the collected data in histograms. Figure 5 depicts such a set of histograms for the time between two successive upsets. Here each sub-graph represents one specific type of upset, as indicated in the respective caption. A data bin is described by the resolution time t_{res} (y-axis) and the time between upsets (TBU, x-axis). The height of the bar (z-axis) gives the relative occurrence of the given time between upsets (relative to others, in percent) for the given resolution time (t_{res}).

It is clearly visible in the figure that for a given resolution time the time between upsets has a pronounced distribution with a considerable spread (i.e. range from the shortest observed time between upsets to the longest). This spread stays approximately independent over resolution time. Note, however, that the time between upsets scaling is logarithmic, so when the mean value moves to the right for higher t_{res} (as, e.g., in sub-graph (b) for falling edges), this actually means the (absolute) spread increases. An important observation is that the spread spans approximately 10 decades. With the mean value being roughly in the middle of this range, this means that we may see time between upsets that are 5 decades shorter than the mean. So with a MTBU of 1000 years we still have a non-negligible probability of seeing an upset within 4 days, and for a reliability analysis it is obviously important to have some qualified estimation for this probability. For example, in the case of rising edges, for t_{res}=471.9ps we have a mean of 3.28s and a minimum observed time between upsets of 0.007s with a frequency of 0.27%).

For $t_{res} < 0$, i.e. when we do not even allow for the nominal output delay, the behavior is distinctly different. As expected, every single output of the UUT is regarded as too late and hence all up- and down-transitions are late.

Notice how the sub-graphs (a) and (c) complement each other: Starting from a t_{res} of zero, up to approximately 127ps we see late up-transitions only, followed by an interval of $127ps < t_{res} < 446ps$ where we do not have a single delayed up-transition but positive pulses instead, as indicated in sub-graph (c). For larger t_{res} we the again observe late up-transitions exclusively. We can witness the same complementary behavior in sub-graphs (b) and (d).

When looking at the resolution time distribution for a fixed time between upsets, one can see a large standard derivation. In other words, a given time between upsets can be observed with many different resolution times. An obvious interpretation would be that this is caused by a detection clock jitter that in turn causes a jitter on t_{res}. This interpretation, however, is invalidated by the sharp boundaries between pulses and late transitions discussed above. A non-negligible clock jitter would blur these boundaries. We therefore conclude that the measured distribution is indeed mainly caused by the metastability behavior and is not an artifact of the measurement circuit.

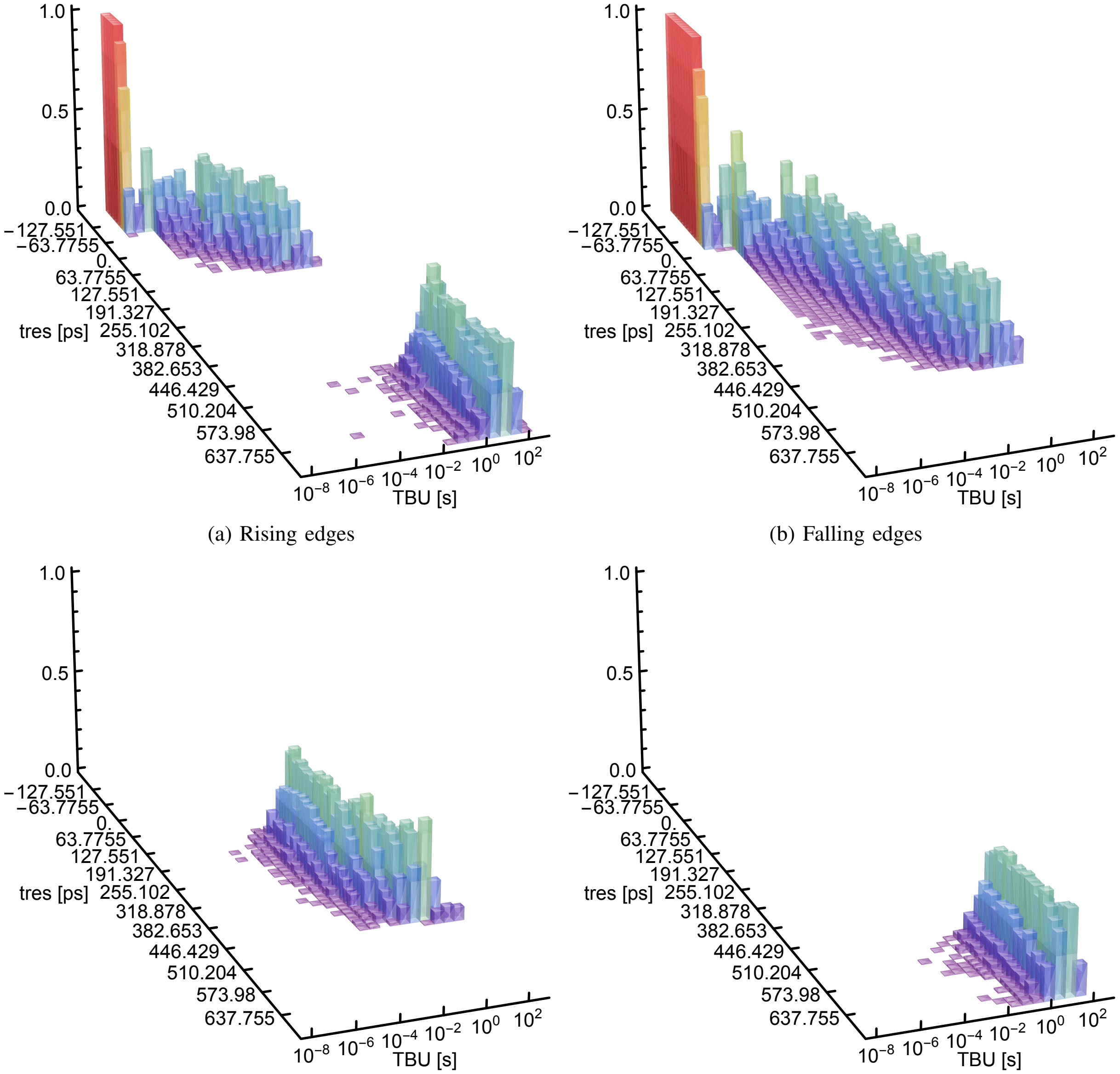
Another, even more compressed, possibility to visualize the results is shown in Figure 6. There a logarithmic box whisker plot is used to depict the results. The box captures 50% of the recorded data while the line within the box corresponds to the mean value of the measurement. The whiskers (the two lines outside the plot) show the range between the minimum and the maximum measured value.

As already mentioned in Section II, the traditional measurement circuits capture the mean value (MTBU) only. In Figure 6 this would exactly correspond to plotting the lines within the boxes. When discarding all the other information we would get a similar results to those from [9] (MTBU is one over the upset rate) and other literature. This illustrates how much more detail our proposed measurement scheme can record.

It is again apparent from the figure that the mean time between upsets and its spread is increasing with increasing resolution time. This figure further shows the differences between master and slave latch quite clearly. While the master latch exhibits positive pulses ($0 \rightarrow 1 \rightarrow 0$), the slave latch produces negative pulses ($1 \rightarrow 0 \rightarrow 1$). The metastability resolution of the slave is much worse than the one for the master (much smaller slope), while exhibiting similar spread.

VI. Conclusion

Starting from the insight that the mean time between upsets is not actually what is needed for a reliability analysis, we have proposed a measurement approach that allows the assessment of the distribution of upsets over time, augmented with a separation of different types of upset, namely late transition (up and down) or glitch (positive or negative). The key extension to state-of-the-art measurement setups is the introduction of a time base that allows time stamping of events rather than counting them. We have illustrated the implementation of this concept on three different FPGA platforms and presented selected results. These results have shown that there is indeed a pronounced distribution of time between upset (TBU) that cannot simply be described as a uniform or normal distribution. Our future research will be directed towards exploring candidate distribution models that fit the observations, and investigating the impact of data and clock signals (correlation, jitter, ...) on the observed distribution. In our experiments so far we have tried to establish a uniform distribution for the relative phase between data transitions and active clock edges (as assumed in the traditional model as well), but it must be further investigated how far small deviations from this ideal setting (periodicity with the GCM of data and clock frequency versus jitter and drift effects) impact the observed TBU distribution. If this turns out to be highly relevant, then it might become necessary to study the relevant parameters in the respective individual application setting as well and consider it in the synchronizer's reliability analysis. Furthermore, once the distribution is identified, a suitable reliability model needs to be established that considers all relevant parameters of the distribution rather than the mean value only.



(a) Rising edges

(b) Falling edges

(c) Positive Pulses

(d) Negative pulses

Figure 5: Histogram for Kintex-7

REFERENCES

[1] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Stanford University, Stanford, CA, USA, Oct. 1984.

[2] L. Kleeman and A. Cantoni, "On the unavoidability of metastable behavior in digital systems," *Computers, IEEE Transactions on*, vol. C-36, no. 1, pp. 109–112, Jan 1987.

[3] D. J. Kinniment, *Synchronization and Arbitration in Digital Systems*. Wiley, 2007.

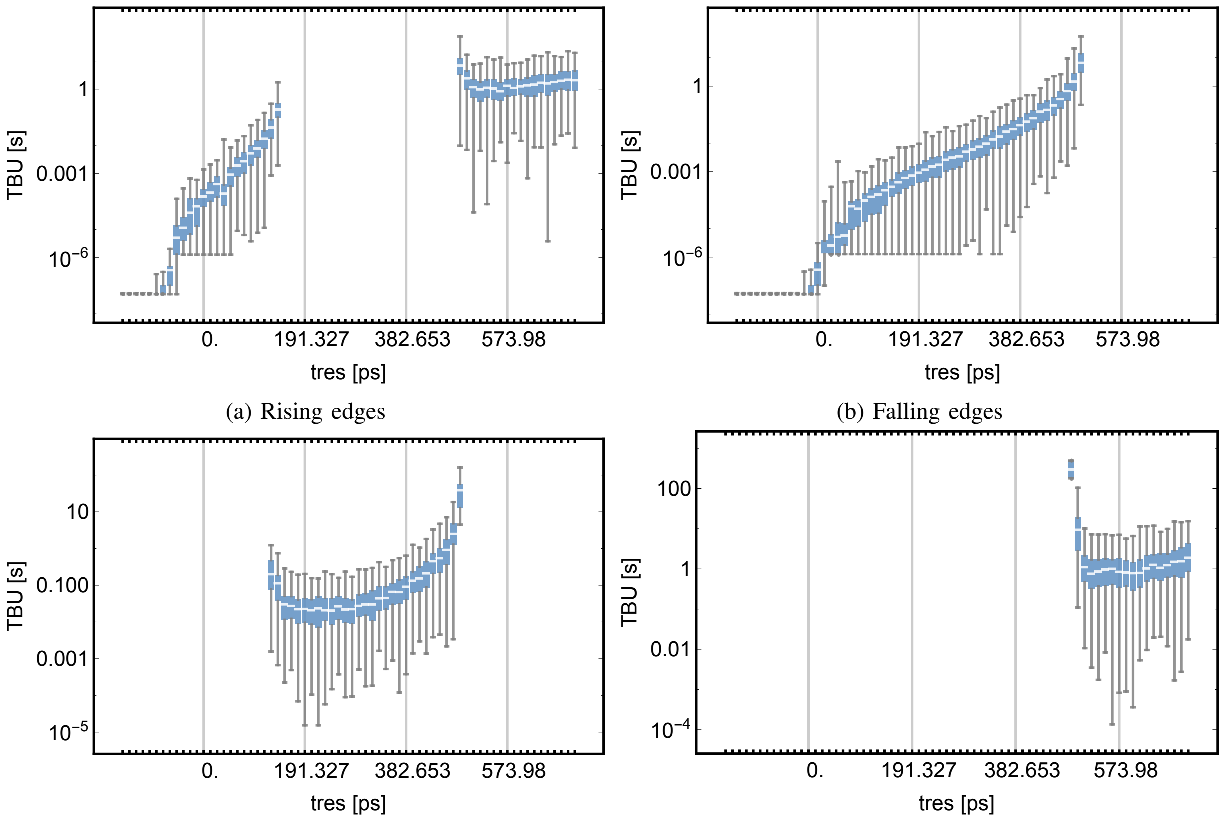
[4] L. R. Marino, "General Theory of Metastable Operation," *Trans. on Computers*, vol. C-30, no. 2, pp. 107–115, 1981.

[5] H. J. M. Veendrick, "The Behavior of Flip-Flops Used as Synchronizers and Prediction of Their Failure Rate," *Journal of Solid-State Circuits*, vol. 15, no. 2, pp. 169–176, 1980.

[6] L. Kleeman and A. Cantoni, "Metastable Behavior in Digital Systems," *Design and Test of Computers*, vol. 4, no. 6, pp. 4–19, 1987.

[7] S. Beer, R. Ginosar, M. Priel, R. Dobkin, and A. Kolodny, "The Devolution of Synchronizers," in *16th IEEE Symposium on Asynchronous Circuits and Systems (ASYNC), 2010*, May 2010, pp. 94–103.

[8] J. Wu, Y. Ma, J. Zhang, Y. Kong, H. Song, and X. Han, "Research on Metastability based on FPGA," in *9th International*



(a) Rising edges

(b) Falling edges

(c) Positive Pulses

(d) Negative pulses

Figure 6: Box whisker plot for Kintex-7

Conference on Electronic Measurement Instruments (ICEMI), 2009, Aug 2009, pp. 4–741–4–745.

[9] T. Polzer and A. Steininger, “An Approach for Efficient Metastability Characterization of FPGAs through the Designer,” in *19th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), 2013*, May 2013, pp. 174–182.

[10] I. Jones, S. Yang, and M. Greenstreet, “Synchronizer Behavior and Analysis,” in *15th IEEE Symposium on Asynchronous Circuits and Systems (ASYNC), 2009*, May 2009, pp. 117–126.