

# The Devolution of Synchronizers

Salomon Beer<sup>1,2</sup>, Ran Ginosar<sup>1</sup>, Michael Priel<sup>2</sup>, Rostislav (Reuven) Dobkin<sup>1</sup> and Avinoam Kolodny<sup>1</sup>

<sup>1</sup>Electrical Engineering Dept, Technion—Israel Institute of Technology, Haifa, Israel

<sup>2</sup>Freescale Semiconductor

{shloimi,michaelpriel}@freescale.com,

{sbeer@tx,dobkin@tx,ran@ee,avinoam@ee}.technion.ac.il

**Abstract**— Synchronizers play a key role in multi-clock domain systems on chip. Traditionally, improvement of synchronization parameters with scaling has been assumed. In particular, the resolution time constant ( $\tau$ ) has been expected to scale proportionally to the gate delay 'FO4'. Recent measurements, however, have yielded counter-examples showing a degradation of  $\tau$  with scaling. In this paper we describe these measurements and validate them with circuit analysis and simulations, demonstrating the devolution of synchronization parameters. Measurements have been made on a 65nm circuit and on series of FPGA devices. The  $\tau$  measured on the 65nm circuit was about 100ps, in contrast with expectations of less than 30ps. Three similar FPGA devices, fabricated in 130, 90 and 65nm processes, yielded values of 57, 51 and 73ps, respectively, showing a significant increase in 65nm relative to older generations. The analysis is validated by simulations that predict further increase of  $\tau$  for future technologies.

**Keywords**- Synchronization, metastability, mean time between failures (MTBF), technology scaling, tau degradation effect, synchronizer degradation.

## 1 INTRODUCTION

This paper presents evidence, analysis and simulations that challenge the common notion that  $\tau$ , the resolution time constant of synchronizers, always scales down with technology.

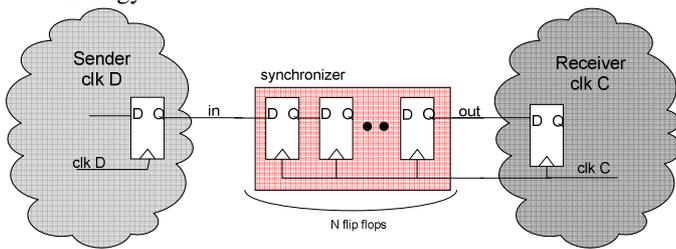


Figure 1: A typical synchronizer using N flip flops

Large multiple-clock domain Systems on Chip (SoC) typically require synchronization when transferring signals and data among the various clock domains and when receiving asynchronous inputs. Such synchronizations are often susceptible to meta-stability effects [1], which may propagate into the receiving circuit and may cause malfunctioning. To mitigate the effects associated with metastability, latches and flip flops are often used to synchronize the data [2], such as the N pipelined flip flops

shown in Figure 1, which reserve a pre-determined time S for metastability resolution,  $S \approx (N-1) \times T_C$  ( $T_C$  is the clock cycle time of the receiving clock domain). There is, however, a finite probability that the circuit will not resolve its metastable state correctly within the allowed time. To enable assessing the risk, and to enable the design of reliable synchronizers and systems, models describing the failure mechanisms for latches and flip flops have been developed [2] [3][4]. Most models express the risk of not resolving metastability in terms of the mean time between failures (MTBF) of the circuit,

$$MTBF = \frac{e^{S/\tau}}{T_w \times F_C \times F_D} \quad (1)$$

where  $F_C$  and  $F_D$  are the receiver and sender frequencies, respectively,  $\tau$  is the resolution time constant, and  $T_w$  is a parameter often related to the setup-and-hold time window at the synchronizer input.

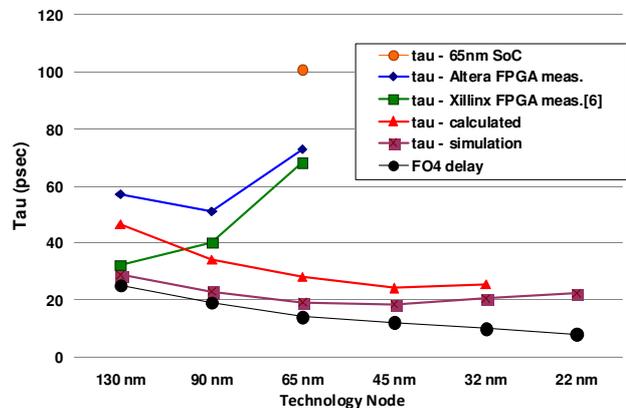


Figure 2: Calculated, measured and simulated  $\tau$

Desirable values of MTBF depend on the application and range from several years upwards. Typical values of  $\tau$  are the same order of magnitude as the gate delay of the technology (often expressed as FO4, the fanout-of-four delay of a standard gate). Evidently, as technology scales, FC and FD increase and to maintain high MTBF (without increasing N)  $\tau$  must decrease as well. Previous works (e.g., [5]) have indeed predicted that  $\tau$  will decrease with technology scaling.

Recent measurement results [6][7], however, have shown that this prediction might be incorrect. As shown in Figure 2 and elaborated below, three different device technologies demonstrate an increase of  $\tau$  with technology scaling, starting towards 90 and 65nm process nodes. Analysis and simulations support these findings for 45nm and beyond. We call this effect *the  $\tau$  degradation of synchronizers*.

The purpose of this work is to provide empirical results showing the degradation of  $\tau$ , to derive an analytic model that can predict these results, and to present simulations that validate the model and confirm the findings. Sect.2 shows measurement results of  $\tau$  for a 65nm SoC circuit and for three FPGA product generations. These measurements are compared with previous measurements done in other technologies [3][5][8]. Sect. 3 develops an analytical model for  $\tau$  that can predict these findings, and Sect. 4 presents simulations showing the degradation of  $\tau$ . Finally, Sect.5 discusses the results and the implications for circuit design and proposes future research.

## 2 EXPERIMENTAL RESULTS SHOWING $\tau$ DEGRADATION.

In this section, measurement evidence of the  $\tau$  degradation effect is presented. Recent measurements provided in [6] using the method described in [9] show a degradation of  $\tau$  with technology scaling for Xilinx Virtex II-pro, IV and V families in 130nm, 90nm and 65nm, respectively. We have measured  $\tau$  values in a 65nm SoC and in three Altera FPGA devices, and have discovered the same trend of  $\tau$  degradation. Figure 2 compares these  $\tau$  measurements with simulations and modeling calculations. Model calculations are explained in Sect. 3. Simulations of  $\tau$  are described in Sect. 4. While our model and simulations indicate  $\tau$  degradation starting at 45nm node, actual measurements reveal such degradation starting earlier, in either 90nm or 65nm nodes.

### 2.1 $\tau$ measurement on a 65nm SoC

A synchronizer test circuit was fabricated as part of a commercial 65nm SoC. The on-chip measurement system,

Figure 3, comprises a shift register that holds the configuration data, an input and clock generation unit (ICG), a design under test unit (DUT) that includes flip flops used as synchronizers, a measuring unit (ME) that includes the measuring circuits of Figure 4, a 16-bit counter and an output serializer.

The measurement consists of sampling the output of the FF-under-test twice: first (X in Figure 4) by the clock delayed by DL (generated by the delay line) and second (Y) by the negative edge of the clock. When using a relatively slow clock (less than 50 MHz), sampling with the negative edge of the clock is considered safe, since it can be assumed that all metastable events have resolved. The two samples are compared by the XOR gate. A metastability event that

resolves during the time window between the delayed and negative edges of the clock (namely an event that did not resolve within the allotted time  $S=DL$ ) increments the counter. The measurement continues for time period T and thus MTBF is T/count. The entire measurement is repeated using different DL delays, obtaining a set of (DL, MTBF) readings. This set is used to compute  $\tau$  according to (1). The method assures higher accuracy and lower measurement noise as compared to methods requiring variable high frequency clocks [9][10]

A controller writes into the shift register in order to configure the DUT and the DL value. The controller sets the measuring period T by enabling and disabling the counter. T can vary from seconds to hours.

Following each measuring period, the controller initiates a serial readout of the counter value. This procedure is repeated for multiple values of DL. The entire test is performed under software control, and readings are further processed by the software.

Measurements were performed at VDD of 1.1V and at room temperature, for several different data frequencies in the range of  $F_D=0.1-8$ MHz, and clock frequency  $F_C=6.25$ MHz. The measurement period T ranged from two minutes to six hours for each DL value in each measurement. Measured values are presented in Figure 5, showing exponential relation of resolution time to the number of events. Two regions were identified, corresponding to short delay (SD) and long delay (LD), for better fit of the exponential parameters, as proposed in [3]. All measurements yielded consistent results for  $\tau$  in the range 96-103psec as shown in Figure 6. Characterization yielded a  $\tau$  value of 100psec with 4% error.

### 2.2 $\tau$ measurement on Altera FPGA devices

This sub-section presents  $\tau$  measurements obtained for three generations of Altera FPGA devices, Stratix (130 nm), Stratix II (90 nm) and Stratix III (65 nm).

The measurements were performed by means of the circuit shown in Figure 7a. Two external clock sources having similar but non-identical frequencies produce DATAIN and CLK signals. The QA flip flop may become metastable when DATAIN and CLK toggle almost simultaneously. The clock for QB flip flop is generated by an on-chip Altera PLL that can produce a programmable phase-shift of CLK.

Thus, QB and QC sample the output of QA at two different times: a phase shift S after the CLK of QA, and at the end of the CLK cycle. QE aligns the output of QB. A metastability event which has resolved in between the two samplings (namely an event that did not resolve within the allotted time S) is detected when QE and QC hold different values. The resolution time S is programmable, and the measurement is repeated for several values of S. This parameter is similar to DL (delay line delay) presented in section 2.1.  $\tau$  is then computed from the empirical data of event count vs. S, similar to Sect. 2.1.

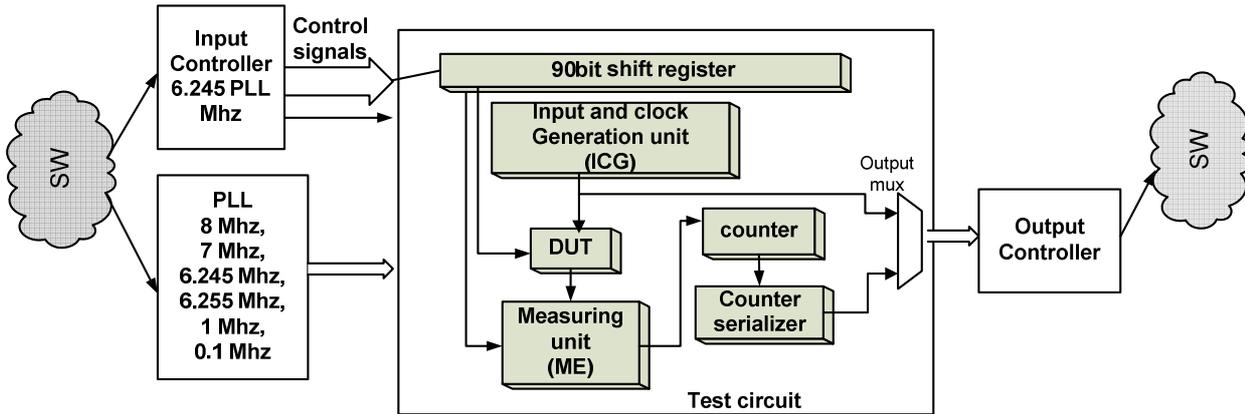
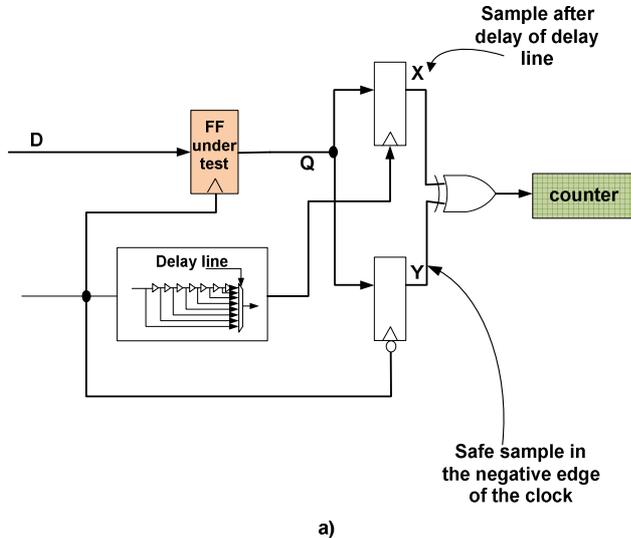
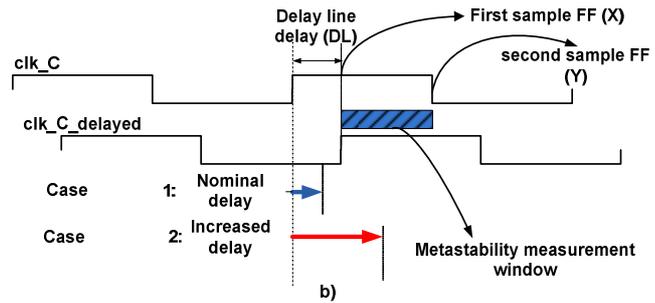


Figure 3: On-chip metastability measurement system



a)



b)

Figure 4: (a) The measurement system (b) Signal waveforms explaining the operation of the test system

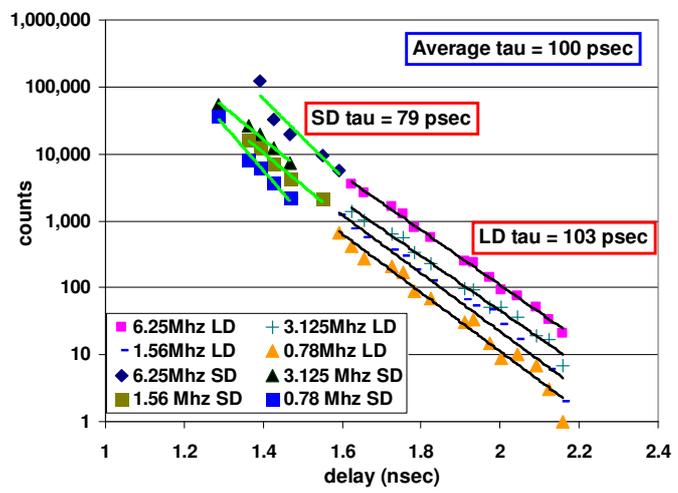


Figure 5: Metastability measurements of a standard library cell FF in the 65nm test circuit.

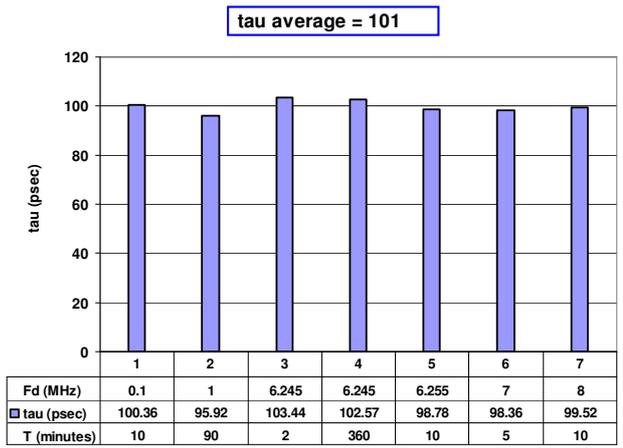


Figure 6: Measured  $\tau$  for different cases

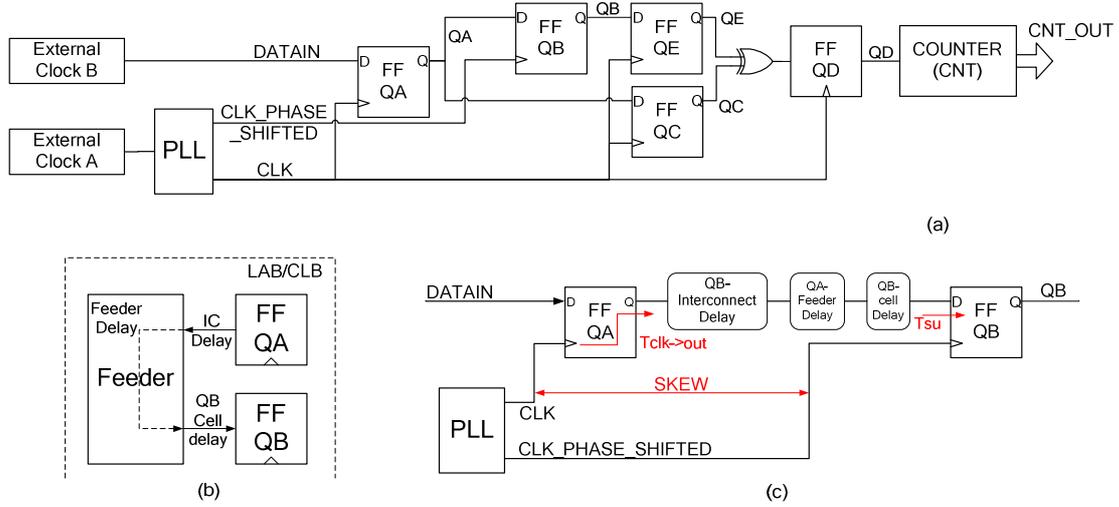


Figure 7: Test circuit for  $\tau$  measurement on Altera FPGAs

The resolution time  $S$  of a two-flip-flop synchronizer is usually assumed to be equal to the cycle time. A more accurate computation of the resolution time should subtract certain propagation delays, such as clock-to-out and setup times of flip flops and interconnect delays. To maximize the resolution time, the two flip flops should be placed close to each other. The interconnect delay problem exacerbates in FPGAs, where the two flip flops are connected through existing interconnect resources. In FPGAs, even when the flip flops reside in the same logic cell (Figure 7b), the combinational and interconnect delays between the synchronizer flip flops are not negligible, especially for high frequency clocking. These delays were controlled in this experiment as follows. The QA and QB flip flops of the test circuit (Figure 7a) were constrained to be placed in the same FPGA cell. The minimal register to register delay (cf. Figure 7c) comprises five components: the clock-to-out delay of QA, the interconnect delay from QA to a 'feeder' cell (an interconnect structure imposed by the Altera architecture), the feeder cell delay, the QB cell delay (interconnect from the feeder to QB) and the QB setup time. All these five delays were kept constant throughout the experiment for all tested devices. Thus, the resolution time  $S$  was computed as follows:

$$S = [SKEW + PHASE\_SHIFT] - [T_{CLK \rightarrow OUT} + D^{IC} + D_{Feeder}^{Cell} + D_{QB}^{Cell} + T_{SU}] \quad (2)$$

The minimal time step of the phase shifts of the Altera FPGA PLL is 179ps (in several exceptions, the time step is 60ps). DATAIN and CLK frequencies were 43.75 MHz and 44 MHz, respectively. The FPGA die temperature was kept at 39°C.

The measurements are presented in Figure 8 and Figure 9. Noting the data for Stratix III, we measured  $\tau = 73$ ps based on the four right-most data points. The left-most data point is not trustworthy, as it appears suspiciously early (too close to the PLL minimal time step). Still, it is shown in the chart for comparison with results published by Altera: The Quartus II synthesis and P&R tool has recently enabled MTBF analysis for Stratix III. Using a simple test case design, the Altera Quartus [10] tool reported MTBF values that (using Eq. (1)) imply  $\tau = 46$ ps,  $T_W = 20$ ps. This result is very close to what can be obtained from Figure 8 if the left-most data point were mistakenly included in the computation, yielding a deceptively low value of  $\tau = 43$  ps.

Figure 9 plots  $\tau$  vs. technology nodes.  $\tau$  scales down when moving from 130nm to 90nm, and increases significantly when moving to 65nm. These results are in agreement with Xilinx FPGA [6], cf. Figure 2.

Incidentally, the measured  $\tau$  values for some special FPGA cells such as IO cells are different than above. This effect demonstrates that  $\tau$  depends on the internal circuit structure of the flip flop.

### 3 MODELING $\tau$

To further understand the anomaly of  $\tau$  degradation, a model describing its behavior is developed. In this paper a voltage model is adopted [11], in order to relate  $\tau$  to FO4 delay, a principal scaling indicator. Correlation to models based on current using  $g_m$  [12][13] can also be shown.

The following analysis relates to a gated latch, comprising two inverters and one transmission gate (TG1) connected in a closed loop, as in Figure 10. We model the gated latch since it is simpler than a complete flip-flop. The two models are closely correlated, when the master and slave latch have the same circuit design.

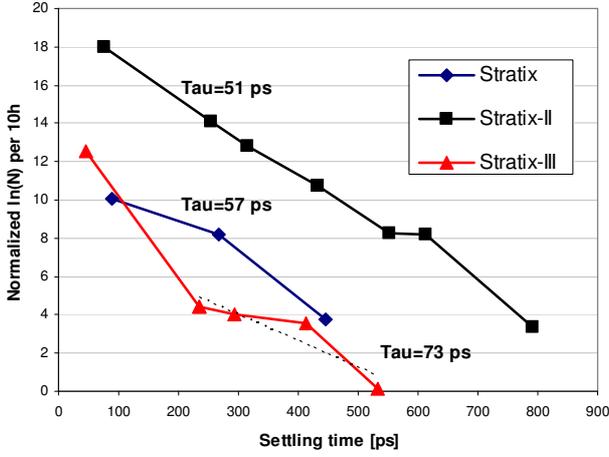


Figure 8:  $\tau$  results for different Altera product generations. Note that for Stratix-III,  $\tau=43$ ps if all measured points are taken into account; however, actually  $\tau=73$ ps since the leftmost point is considered a measurement error.

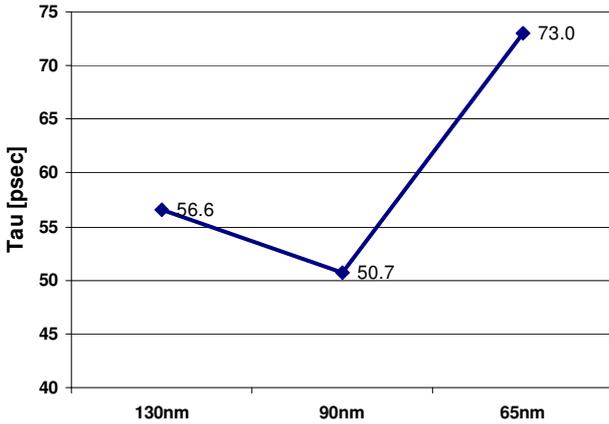


Figure 9: Measured  $\tau$  vs. technology in Altera FPGA

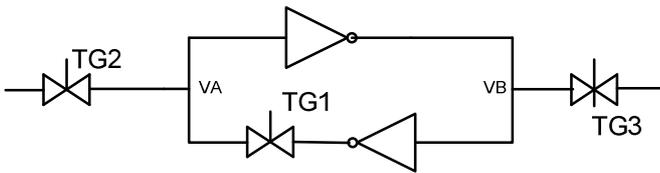


Figure 10: Circuit of a gated latch

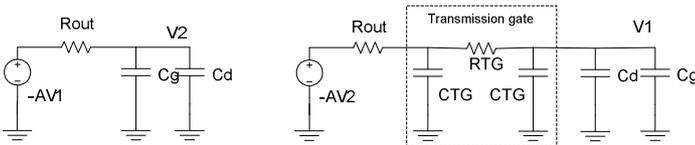


Figure 11: Model of a gated latch

Each inverter is modeled as a voltage source (Figure 11), feeding the gate oxide capacitance of the transistors of the other inverter and the drain capacitance of the transmission

gates (TG2, TG3). The transmission gate in the loop (TG1) is modeled as a  $\pi$  network. The two inverters are assumed identical.

$V_1$  and  $V_2$  are the voltages on the input of the inverters,  $R_{out}$ ,  $A$  and  $C_g$  are the output resistance, the voltage gain and the gate input capacitance of the inverters operating in the saturation region, and  $C_d$  is the diffusion capacitance of the transmission gates connected to the nodes as shown in Figure 11. This scheme can be simplified by changing the representation of TG1 from a  $\pi$  to a  $r$  model yielding the simplified circuit of Figure 12, in which:

$$\begin{aligned} R_1 &= R_{out}, C_1 = C_g + C_d \\ R_2 &= R_{out} + R_{TG}, C_2 = C_g + C_d + C_{TG} \end{aligned} \quad (3)$$

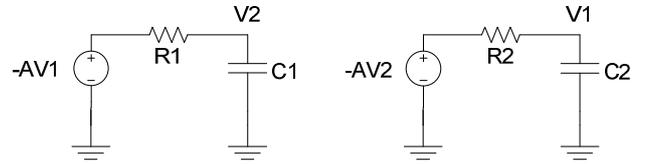


Figure 12: Simplified model of the gated latch

Following [11], the equations describing the behavior of the system can be written as:

$$\tau_1 \tau_2 \frac{d^2 V_1}{dt^2} + \frac{(\tau_1 + \tau_2)}{A} \frac{dV_1}{dt} + \left( \frac{1}{A^2} - 1 \right) V_1 = 0 \quad (4)$$

where:

$$\tau_1 = \frac{C_1 R_1}{A}, \quad \tau_2 = \frac{C_2 R_2}{A} \quad (5)$$

These two constants may be different from each other depending on the feedback characteristics of the latch and the TG properties. The solution of (4) is

$$V_1(t) = K_1 e^{t/\tau_a} + K_2 e^{t/\tau_b} \quad (6)$$

The constants  $\tau_a, \tau_b$  are obtained by substituting (6) in (4), giving:

$$\tau_{a,b}^{-1} = \frac{1}{2\tau_1 \tau_2} \left( -\frac{\tau_1 + \tau_2}{2} \pm \sqrt{\left( \frac{\tau_1 + \tau_2}{2} \right)^2 - 4\tau_1 \tau_2 \left( \frac{1}{A^2} - 1 \right)} \right) \quad (7)$$

Assuming that the output resistance of the transistor has the same value as the resistance of TG1, then  $R_{out} = R_{TG}$ ,  $C_{TG} = C_g + C_d$ . Substituting these values in (5) gives  $\tau_2 \approx 4\tau_1$ , and substituting in (7) yields:

$$\tau_a = \frac{16}{3} \tau_1, \quad \tau_b = \frac{-16}{13} \tau_1 \quad (8)$$

Equations (6) and (8) show that the exponent is proportional to  $1/\tau_1$ . The magnitudes of the constants multiplying  $\tau_1$  in (8) decrease when the transmission gate grows larger relative to the inverters size; we discuss below the implication of this observation. We can ignore the negative exponent  $\tau_b$  (following [2], [3], [11]) since its effect is limited to a very short time after clock triggering. We define  $\tau_a = \tau$  and rewrite:

$$V_1(t) = K_1 e^{t/\tau} \quad (9)$$

Next, consider the dependence of  $\tau$  on technology scaling. We wish to express  $\tau$  by means of FO4 delay,  $t_{d,FO4}$ . Assuming that  $t_{d,FO4} = 4 \cdot t_{d,FO1}$ , then:

$$\frac{t_{d,FO4}}{4} = R_{average} \cdot C \quad (10)$$

where  $C$  is the input capacitance of one inverter and  $R_{average}$  is the effective (average) resistance of the transistor, when the transistor discharges (charges) a capacitance. Observe that  $R_{average}$  in (10) is not the same as the resistance in (5), which relates to the saturation mode ( $R_{sat}$ ). Following [14]:

$$R_{average} = \frac{1}{-V_{DD}/2} \int_{V_{DD}}^{V_{DD}/2} \frac{V}{I_{DSAT} \cdot (1 + \lambda V)} dV \quad (11)$$

$$\approx \frac{3}{4} \frac{V_{DD}}{I_{DSAT}} \left( 1 - \frac{7}{9} \lambda V_{DD} \right)$$

with

$$I_{DSAT}(V_{GS})|_{V_{DD}} = C_{ox} \mu \frac{W}{L} \times \left( (V_{DD} - |V_t|) V_{DSAT} - \frac{V_{DSAT}^2}{2} \right) \quad (12)$$

where  $\lambda$  is the parameter of channel length modulation. Turning to  $R_{sat}$ , the output resistance when the transistor is in metastability is given by [15]:

$$R_{sat} = \frac{1}{2 \lambda \cdot I_{DSAT}} \quad (13)$$

with:

$$I_{DSAT}(V_{GS})|_{V_{DD}/2} = C_{ox} \mu \frac{W}{L} \left( \left( \frac{V_{DD}}{2} - |V_t| \right) V_{DSAT} - \frac{V_{DSAT}^2}{2} \right) \quad (14)$$

In order to express (5) by means of FO4 delay we compute the ratio between (13) and (11) giving:

$$\frac{R_{sat}}{R_{average}} \approx \frac{2}{3} \frac{1}{\lambda \cdot V_{DD}} \left( \frac{V_{DD} - V_T}{\frac{V_{DD}}{2} - V_T} \right) \triangleq \eta \quad (15)$$

Then, (5) can be rewritten as

$$\tau = \frac{R_{sat} \cdot C}{A} = \frac{C \cdot \eta \cdot R_{average}}{A} = \frac{\eta}{4} \frac{t_{d,FO4}}{A} \quad (16)$$

Thus,  $\tau$  can be expressed as a function of the inverter delay, the inverter gain,  $V_{DD}$ ,  $V_T$  and  $\lambda$ . This approach is also validated by simulations in section 4. In the past when  $V_{DD} \gg V_T$ , the value for  $\tau$  ranged from 0.2 to 1.5 FO4 delays, matching the rules of thumb presented in previous works [4][16]. The purpose now is to investigate (16) to gain insight into how  $\tau$  evolves with technology scaling. This is explained in the following section.

## 4 SIMULATION RESULTS

In this section we present two types of simulations: *Indirect simulations*, where each one of the components of (16) is simulated and then combined to form  $\tau$ , and *direct simulations* where  $\tau$  is calculated from a latch resolving from metastability using the method employed in [3]. Alternative simulation methods (such as [11][16][17]) may also be employed to investigate this phenomenon.

### 4.1 Indirect simulations

Circuit simulations using SPECTRE were carried out in order to validate the model of Sect. III. Simulations of FO4 delay, gain  $A$  and the  $\eta$  parameter, the three components affecting  $\tau$  in (16), are presented. PTM parameters [18] for bulk processes were used, employing BSIM3 ([19]) level 49 models for 180—130nm and BSIM4 ([20]) level 54 models for 90—22nm devices.  $V_{DD}$  values per technology nodes follow the ITRS [21][22].

#### 4.1.1 Delay simulations

Simulated FO4 delay and gain vs. technology nodes are shown in Table 1. The results are consistent with previous work [23], and are presented here as a proof of simulation consistency. Simulated inverters had minimal channel length, fixed width  $W$  and an aspect ratio of two. Figure 13 illustrates FO4 delay vs. technology nodes, showing the expected decrease in delay with scaling.

#### 4.1.2 Gain simulations

Simulations of inverter gain  $A$  were carried out using two methods: calculating the derivative of the voltage transfer curve (VTC) of the inverter in the balance (metastability) point, and simulating the gain of the inverter as an amplifier, biased in the balance point. The two methods yielded consistent results. Figure 14 shows the VTC (top) and its derivative (bottom) for various  $V_{DD}$  values for a 130nm process.

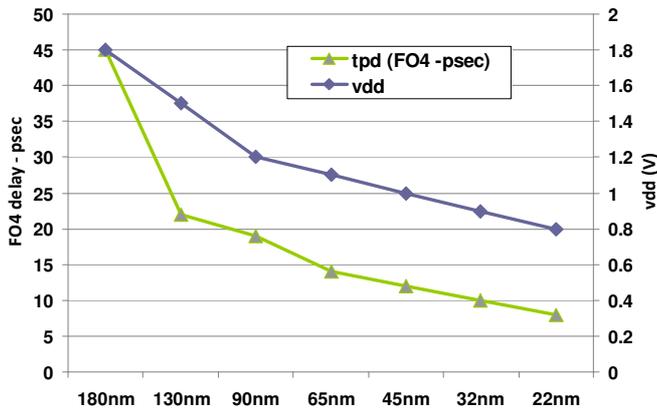


Figure 13: FO4 delay simulation vs. technology node

Table 1: Delay and gain simulations

Technology node	V <sub>DD</sub> (V)	V <sub>T</sub> (V)	FO4 (psec)	Gain (A)
180 nm	1.8	0.40	45	7.41
130 nm	1.5	0.33	22	8.06
90 nm	1.2	0.26	19	7.12
65 nm	1.2	0.17	14	6.16
45 nm	1.0	0.15	12	4.78
32 nm	0.9	0.14	10	3.59
22 nm	0.8	0.14	9	2.59

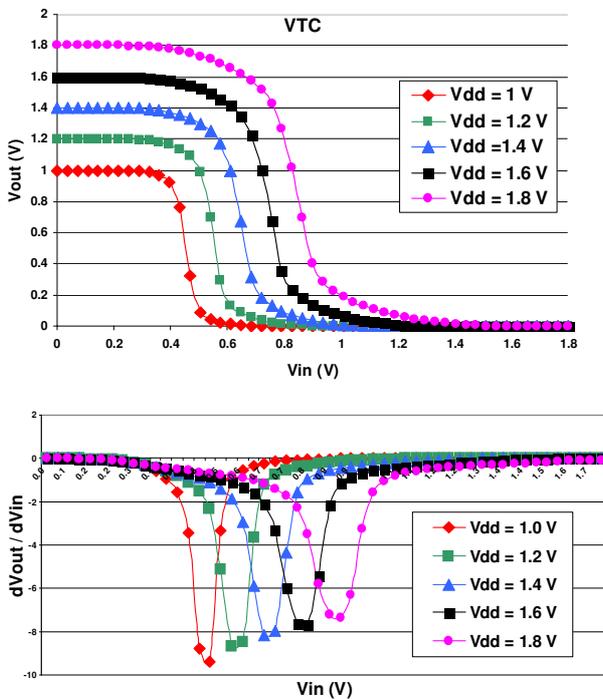


Figure 14: VTC and VTC derivative of a 130nm inverter

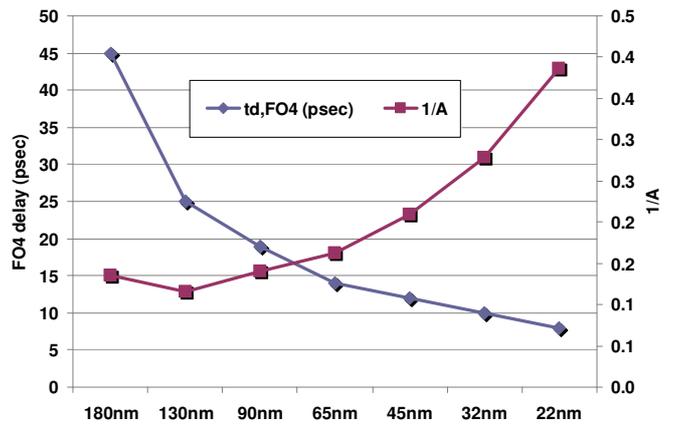


Figure 15: FO4 delay and 1/A simulations

Figure 15 presents the scaling of FO4 delay and the latch gain (inverted, 1/A). The gain appears to decrease faster than FO4 with scaling, partly explaining the increase of  $\tau$  in future technologies.

Figure 16 charts the product of all components comprising  $\tau$ , contrasted with FO4 delay. While FO4 delay decreases with scaling, the metastability resolution time constant  $\tau$  decreases for longer channels but the trend reverses and it increases for advanced process nodes, generating an inflection point around 65nm. Thus, the 'traditional' rule of thumb for  $\tau$  of  $0.2\text{--}1.5 \times \text{FO4}$  gate delays, which used to be appropriate for older technologies [4][16], is no longer valid, in light of the  $\tau$  degradation effect.

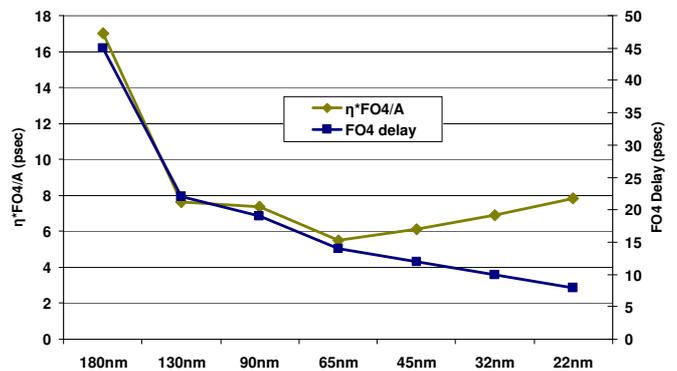


Figure 16: Evolution of factors of  $\tau$ , compared with the evolution of FO4 delay.

#### 4.2 Direct $\tau$ simulations

To corroborate the previous results, direct  $\tau$  simulations were performed, following the method used in [3], as in Figure 17. A switch is connected between the two nodes of the latch to force the latch into metastability where  $V_A$  almost equals  $V_B$  (if they were exactly equal, the simulator

would be unable to resolve). The switch is then released and the latch resolves to a stable state.

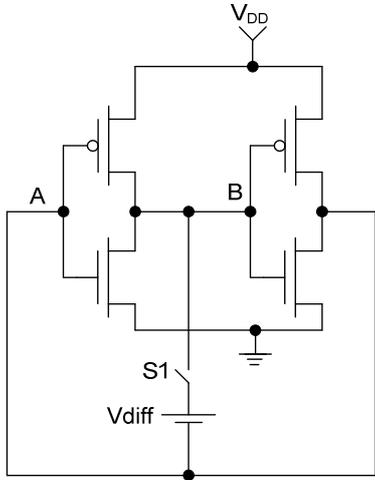


Figure 17: Circuit for direct simulation of  $\tau$

As shown above, the voltages  $V_A$ ,  $V_B$  grow exponentially with time constant  $\tau$ . According to (9),

$$\frac{d}{dt} \ln V = \frac{1}{\tau} \quad (17)$$

Thus,  $\tau$  can be extracted from the slope of the logarithm of the node voltages. Table 2 lists the results for the direct and indirect simulations, and also includes the ratio  $\kappa \triangleq \tau/FO4$ , which represents the 'traditional' rule of thumb.

Figure 18 plots the results for the direct and indirect simulations of  $\tau$ . We see that the tendency of increasing  $\tau$  after an inflection point is consistent. Recall Figure 2 comparing these simulations with measurements of actual circuits.

Figure 19 plots the  $\kappa$  ratio. It is evident that  $\kappa$  was almost constant for technologies older than 65nm. However, it rises with scaling beyond the 65nm inflection point.

To conclude this section, Figure 20 presents scaled and normalized values of  $\tau$  and FO4 delay vs. technology nodes. The  $\tau$  values are the average of the two simulations of Table 2 and are the same as in Figure 2. Clearly, the data indicates a widening gap between  $\tau$  and FO4 delays due to the  $\tau$  degradation effect.

Table 2: Direct and indirect  $\tau$  simulations

Tech. node	FO4 (psec)	$\tau_{direct}$ (psec)	$\tau_{indirect}$ (psec)	$\kappa = \tau/FO4$
180 nm	45	17.02	17.01	0.38
130 nm	22	9.07	7.63	0.41
90 nm	19	7.22	7.38	0.39
65 nm	14	6.02	5.52	0.43
45 nm	12	5.78	6.10	0.48
32 nm	10	6.46	6.90	0.65
22 nm	9	7.03	7.81	0.88

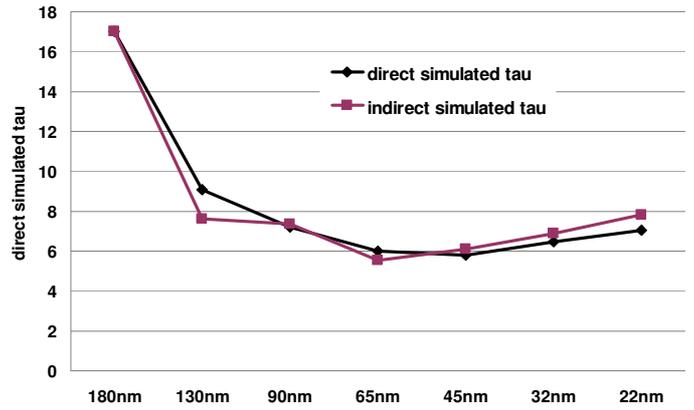


Figure 18: Direct and indirect simulation of  $\tau$

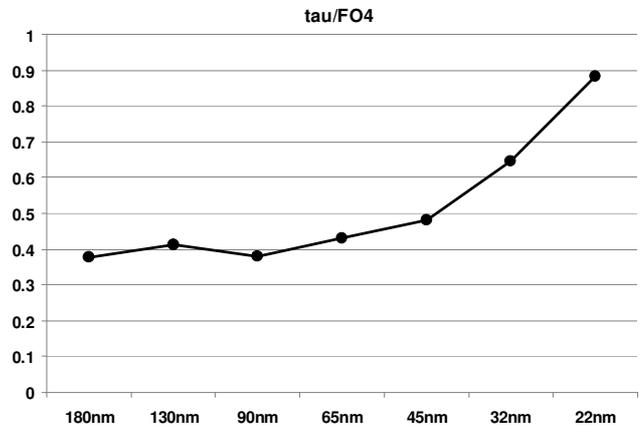


Figure 19: Simulation of  $\kappa = \tau/FO4$

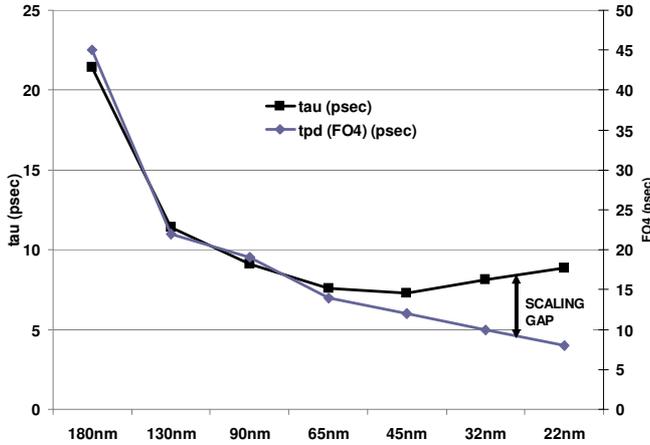


Figure 20: The  $\tau$  degradation effect

## 5 DISCUSSION AND CONCLUSIONS

We have discussed the behavior of latches used as synchronizers. The voltage model was extended to derive  $\tau$  as a function of delay, gain, supply voltage, threshold voltage and the channel length modulation factor. Simulations of these quantities versus scaling reveal that  $\tau$  decreases until an inflection point around 45nm, and thereafter its value increases with further scaling. Other simulations, observing a latch as it resolves metastability, further confirmed the same results. We have called this effect *the degradation of  $\tau$  with technology scaling*.

We have presented evidence of the  $\tau$  degradation effect in actual synchronizers. Measurements of a 65nm circuit in LP bulk CMOS gave  $\tau=100\text{ps}$ , significantly higher than expected for a 65nm device. Corroborating results were measured on Altera FPGA devices that showed a degraded  $\tau$  in the 65nm node relative to 90nm and 130nm. Similar effects were also found on Xilinx FPGA devices [6].

While simulations and the study of scaling effects, as presented in this paper, have predicted the inflection point around the 45nm node, it was actually demonstrated at 90nm and 65nm in measurements of real circuits. This difference may be explained by the difference in circuit design of flip flops in different devices and in our simulations. This difference can also be explained by the fact that we have conducted our measurements in flip flops while our model relates to the gated latch circuit. The  $\tau$  for the whole flip flop may differ from the  $\tau$  of the master and slave latches, and this difference is intrinsically related to the circuit design of the latches. Usually library flip flops, as the ones we used in FPGA and SoC measurements, are designed with different aspect ratios between master and slave latch (in order to decrease its delay), and thus having different  $\tau$  values for the master and slave latch, which may lead to underestimated values of failures rates compared to the latch model, as shown in [16].

Interestingly, several previous reports of analysis and measurements of synchronizers, although based on different latch designs, different methods and different technologies, produced results that were quite compatible, as well as supporting the decrease of  $\tau$  with scaling: [5] reported  $\tau$  of 78ps, 54ps and 22ps for 0.35 $\mu\text{m}$ , 0.25 $\mu\text{m}$  and 0.18 $\mu\text{m}$  CMOS, respectively (cf. our simulations of  $\tau = 17\text{ps}$  for 0.18 $\mu\text{m}$ ). [3] reported  $\tau = 20\text{ps}$  for 0.25 $\mu\text{m}$ , and [8], following [24], reported  $\tau = 27\text{ps}$  for 0.18 $\mu\text{m}$ . These results evidently validate our own simulations and methods, and also confirm that for a 65nm circuit we would have expected a much lower  $\tau$  compared to the 100ps that we actually measured, demonstrating the  $\tau$  degradation effect.

Process *flavors* (GP, LP, HVT, LVT, *etc.*), are also an important factor when measuring  $\tau$  for a given process. These parameters affect the terms in (16): FO4 delay for a LP process is longer than for GP, resulting in longer  $\tau$ . Also, different  $V_T$  (HVT, LVT) affect the Gain (A) and FO4 terms in (16).

Traditionally, a simple rule of thumb was used for designing synchronizers, stating that  $\tau = \kappa \cdot t_{d,FO4}$  for some fixed  $\kappa$ . In light of the findings of the  $\tau$  degradation effect presented in this paper, as demonstrated in Figure 19, that rule should be questioned in future technology nodes. While a conservative approach of  $\tau = 2 \cdot t_{d,FO4}$ , may still be valid for certain situations, we suspect that in the future each FF and latch design in each technology node and each process flavor should be carefully measured and characterized. In addition, process variations influence circuit performance.  $\tau$  Standard deviation due to process variations in 45nm node is near 50% ([25]), which means a safety parameter should be added to account for worst case scenarios, leading to  $\kappa \approx 3$ .

As an example of the implications of the  $\tau$  degradation effect and of the danger in using the traditional rule of thumb, we consider the number of synchronization stages needed in a typical 65nm circuit. Two values of  $\tau$  are employed: the one we measured on an LP process (100ps) and the one predicted by the rule of thumb  $\tau \approx 2 \cdot t_{d,FO4} = 56\text{ps}$ <sup>1</sup>. Consider synchronizing a data signal of  $F_D=150\text{ MHz}$  into a clock domain of  $F_C=300\text{ MHz}$  and desiring MTBF longer than one million years. Only two flip flops are needed when relying on  $\tau=56\text{ps}$  (enabling a mere single clock cycle of resolution time), while three flip flops must be used if  $\tau=100\text{ps}$  is adopted (Figure 21). The two flip flop design would provide a much lower MBTF in that case: only 3 days.

<sup>1</sup> The simulated FO4 delay of 14ps (Table 1) relates to GP 65nm process; to compare with a measured  $\tau$  on LP process we assume conservatively a double delay, 28ps.

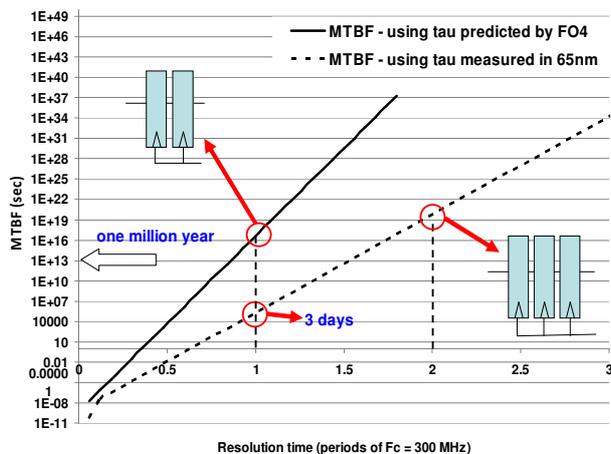


Figure 21: MTBF of synchronizers using different  $\tau$  values.

The value of  $\tau$  is not merely an inherent technology parameter: as is evident from the analysis conducted in Sect. 3,  $\tau$  also depends on the specific circuit used for the synchronizing latch. It may be possible to mitigate part of the  $\tau$  degradation effect by careful circuit design. For instance, it was noted in Sect. 3 (Eq. (8)) that sizing the transmission gate larger relative to the inverters may help to improve  $\tau$ . Other circuit techniques may also be investigated.

## 6 ACKNOWLEDGMENT

The authors wish to thank Peter Alfke for illuminating discussions and his contributions of metastability data for Xilinx devices, Gidel Inc. for their support of the measurement of Altera devices, and Freescale Semiconductor for allowing the inclusion of the test circuit in one of their 65nm SoC.

## 7 REFERENCES

- [1] T.J. Chaney and C.E. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Trans. Comp.*, 22:421-422, 1973.
- [2] L. Kleeman and A. Cantoni, "Metastable behavior in Digital Systems," *IEEE Design & Test of Computers*, 4(6):4-19, 1987.
- [3] C. Dike and E. Burton, "Miller and noise effects in synchronizing flip-flop" *IEEE JSSC*, 34(6):849-855, 1999.
- [4] R. Ginosar: Fourteen ways to fool your synchronizer, *Proc. ASYNC*, 2003.
- [5] M.S. Baghini, M.P. Desai, "Impact of technology scaling on Metastability performance of CMOS synchronizing latches", *Proc. ASP-DAC/VLSI* pp. 317-22, 2002.

- [6] P. Alfke, <http://forums.xilinx.com/t5/PLD-Blog/Metastable-Delay-in-Virtex-FPGAs>.
- [7] S. Beer, R. Dobkin and R. Ginosar, "Metastability measurements of several ASIC and FPGA synchronizers," Technical Report, EE Dept, Technion, Oct. 2009.
- [8] J. Zhou, D.J. Kinniment, G. Russell, and A.V. Yakovlev, "On-Chip Measurement of Deep Metastability in Synchronizers," *IEEE JSSC*, 43(2), 2008.
- [9] P. Alfke. Metastability considerations, XILINX application note XAPP077, 1997.
- [10] J. Stephenson et al., "Understanding metastability in FPGA's," Altera WP-01082-1.1, 2009.
- [11] D.J. Kinniment, A. Bystrov and A.V. Yakovlev, "Synchronization circuit performance," *IEEE JSSC*, 37(2):202-209, 2002.
- [12] C.L. Portmann and T.H.Y. Meng, "Metastability in CMOS Library Elements in Reduced Supply and Technology Scaled Application". *IEEE JSSC*, 30(1), 1995.
- [13] L.S. Kim and R.W. Dutton, "Metastability of CMOS latch/flip-flop", *IEEE JSSC*, 25(4):942-951, 1990.
- [14] J. Rabaey et al., "Digital Integrated Circuits: A Design Perspective," 2nd. ed., Prentice Hall Publishers, Inc., 2003.
- [15] Y. Taur and T.H. Ning, "Fundamentals of Modern VLSI Devices," Cambridge Univ. Press, 1998.
- [16] I.W. Jones, S. Yang and M. Greenstreet, "Synchronizer Behavior and Analysis," *Proc. ASYNC*, 2009.
- [17] S. Yang and M. Greenstreet, "Computing synchronizer failure probabilities," *Proc. Design, Automation and Test in Europe (DATE)*, 2007.
- [18] PTM, Predictive technology model, <http://www.eas.asu.edu/~ptm>.
- [19] BSIM3 Manual, Univ. California, Berkeley, CA, 2000.
- [20] BSIM4 Manual, Univ. California, Berkeley, CA, 2005.
- [21] International Technology Roadmap for Semiconductors (ITRS), 2006 report update.
- [22] The International Technology Roadmap for Semiconductors (ITRS), 2005.
- [23] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm early design exploration," *IEEE Trans. Elect. Dev.*, 53(11):2816--2823, 2006.
- [24] D. Kinniment, K. Heron and G. Russell, "Measuring Deep Metastability", *Proc. ASYNC*, pp. 2-11, 2006.
- [25] J.Zhou, D.Kinniment, G.Russell, and A. Yakovlev, "Adapting synchronizers to the effects of on chip variability", *Proc. ASYNC*, 2008.