

Verification of Function Stable Muller C-element in FPGA

Florian Deeg, Florian Eiermann, Sebastian M. Sattler
Chair of Reliable Circuits and Systems
Friedrich-Alexander-University Erlangen-Nuremberg
Paul-Gordan-Str. 5, 91052 Erlangen, Germany
Email: {florian.deeg,florian.eiermann,sebastian.sattler}@fau.de

Abstract—The goal of the paper is to design a function stable Muller C-element in Field Programmable Gate Array (FPGA). It is a prerequisite for correct asynchronous designs in FPGA [8, 10]. The trick is to provide an easy-to-understand measurement and test to guarantee the correct behavior of the circuitry. The feedback path of the circuitry must have a smaller delay than the forward path of the circuitry. The paper deals with path delays in FPGA at low level, their determination by oscillation and a test for verification. The knowledge gained is intended to serve the asynchronous design process.

Keywords - asynchronous feedback, FPGA, functional safety, stabilization, Muller C-element

I. ORGANIZATION OF THE PAPER

After a brief motivation of the work, the three types of stabilization are discussed in general. Then, the look-up table (LUT) as the basic logic device in FPGA as well as the implementation of the Muller-C element in such a LUT with the output being feedback to the input is presented. Hereafter, the input corresponding to the smallest feedback loop path delay (FLPD) is determined through measurement with a LUT oscillating due to the feedback loop and an asynchronous counter reducing the frequency to be within the bandwidth of the available oscilloscope. The function stable implementation of the Muller-C element is verified through a test. Finally, a conclusion of the measured data is given and the remaining work is addressed.

II. MOTIVATION

ALTHOUGH synchronous circuitry is the more advanced technology, considering asynchronous circuitry with various advantages (lower power consumption, better system performance and no clock skew problems) has become increasingly important in recent years [13]. Synchronous circuits adjust their clock to the worst case path delay to allow the system enough time to process and the clock signal is a single-point of failure of the circuit. In order to achieve higher system performance and higher security, the obvious goal is to avoid the global clock by designing asynchronous circuits. But for this it is important to know and balance the delay times of the used circuit and thus realize stable circuits, otherwise errors like hazards and races can occur. FPGAs are becoming a good choice for prototyping circuits, as these standard

parts have proven to be extremely useful for application-specific integrated circuit (ASIC) verification due to their programmability, availability, and associated electronic design automation (EDA) tools. In addition, FPGAs have evolved to be not just a prototype, but the final realization of hardware, called an accelerator and can offer new opportunities in the automotive sector by allowing hardware to be reconfigured and by enabling faster hardware processing, for example through parallelism, compared to software solutions [4]. However, to use them for asynchronous circuits, standard processes need to be adapted.

III. STABILIZATION TYPES

In the following, different stabilization types are briefly explained to understand how the Muller-C element can be stabilized in the FPGA.

A. State Stabilization

The state stabilization is done with the RS buffer (RSB) [15]. The circuit diagram at transistor level (TL) is shown in figure 1. First the input signals s and r are considered. If the signals

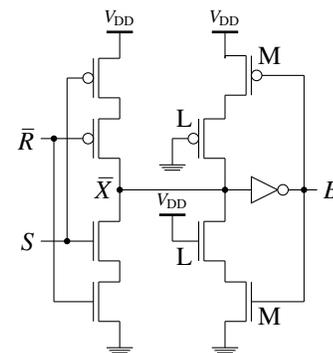


Fig. 1: RS Buffer in TL

are inverted to each other, the tristate is switched, with $(s, r) = [10]$ the output $b = 1$ as well as for $(s, r) = [01]$ the output $b = 0$. If there is an input signal $s \sim r$, i.e. s and r are equivalent to each other, when looking at the output of the tristate of node \bar{X} , High-Z is seen and the babysitter keeps the old state.

Thus, the output pin B is isolated from the input signals. Eq. 1 represents a part of the babysitter.

$$^a z = \delta(^a z, *) \quad (1)$$

The state stabilization has an effect in that firstly, if the energy level of the input signal is too low (i.e. setup and hold times were not met), the tristate does not switch. Secondly, if there are path delays, i.e. that $s \sim r$ applies for a short time, then the tristate switches to High-Z and the old state is held.

B. Function Stabilization

Function stabilization means that each incoming edge (edge event) is stabilized by itself [17, 16] (same edge event) (graphically, a self loop can be drawn). This node is then called a function stabilized state. This self loop must actually be implemented. The state stabilization is axiomatic, therefore it does not have to be drawn in the SFG at all. The self loop of the function stabilization must be specified by a self loop. The edge \bar{A} leads from the node Z_0 into the state \bar{Z}_0 , since

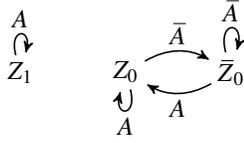


Fig. 2: SFG of a function stabilized Automaton (additional a function stabilized isolated knode)

the latter has the same edge as a self loop. The same applies for edge A from state Z_0 into node Z_0 . Expressed in formulas, the condition for function stabilization (in 2nd Order Logic) is given in Eq. 2.

$$^a z = \delta(^a z, x) \quad (2)$$

where z is the state and $x = (a)$.

C. Structure Stabilization

In structure stabilization, the circuit is supplemented by local stabilization, e.g. memory elements, and stabilized by buffering critical paths of the circuit and triggering them by a clock signal [6]. Thus, a structure stabilized circuit is a circuit synchronized by an applied clock signal, and the structure stabilization is indicated by a circle in the SFG. The formula for structure stabilization is given in Eq. 3.

$$^a z := \delta(^a z, x) \quad (3)$$

IV. LOOK-UP TABLE

Since we cannot implement state stabilization in the FPGA, due to the lack of a high-Z, and structure stabilization contradicts the asynchronous design philosophy due to the clock, a function stable Muller C-element is designed, see Fig. 3. The corresponding truth table of the Muller C-element is given in Tab. I. To implement the Muller C, a LUT, the logical component of the FPGA, is used with the output being fed

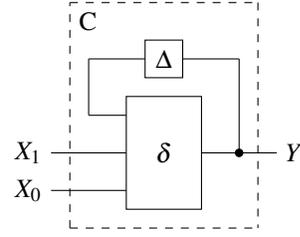


Fig. 3: Muller C-element

TABLE I: Truth Table of the Muller C-element

X_1	X_0	Y	Comment
0	0	0	Reset
0	1	Y	Hold
1	0	Y	Hold
1	1	1	Set

back to the input [12]. For the Muller C to be safe and function stable, the forward path must have a higher delay than the feedback path, the condition for the Muller C in Fig. 3 must therefore be $\tau_\delta > \tau_\Delta$ [7]. The multiplexer structure in TL of a LUT [5] will be briefly discussed, see Fig. 4. The structure

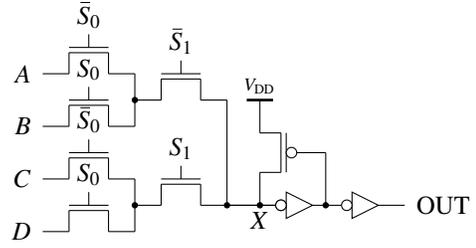


Fig. 4: Two-Stage MUX using Pass Transistors

consists of the input pins S_1 and S_0 , which are applied in such a way that exactly one path is active at a time. Since an NMOS cannot drive a perfect 1 (V_{DD}), a so-called level restorer is attached to the point X to pull the signal up to V_{DD} . We now consider a logic-level multiplexer structure that has six inputs to realize a truth table with 2^6 entries, see Figure 5 [3]. This structure corresponds to the 6-input LUT of the

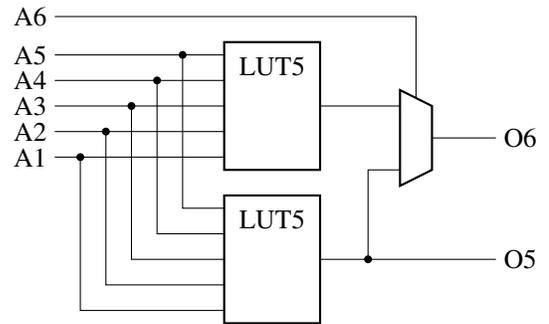


Fig. 5: Structure of a 6-input LUT in Xilinx 7 Series FPGAs

Xilinx 7 series. It has two different outputs O_5 and O_6 . From the structural observation in Fig. 4, it is expected that using input A6 for feedback should result in the lowest FLPD, as input A6 is closest to the output.

V. MEASUREMENT OF THE FEEDBACK LOOP PATH DELAY

The FLPD is equal to the sum of the forward path delay τ_δ and the feedback path delay τ_Δ and depends on the LUT input pin used. To ensure function stabilization and to maximize the switching speed of the Muller C-element, the input of a LUT corresponding to the smallest FLPD is determined by measurement.

A. T-Buffer

The main component of the measurement is the T-Buffer (TB) implemented in a LUT. The output B of the LUT is fed back to the input so that the new output state depends on the old output state. When the LUT is triggered, the new output state is set to the switched value of the old output state. As a result, the TB has an oscillating state with a period equal to twice the FLPD. The period or frequency of the oscillation varies depending on which input pin is used to feedback the output. The truth table for the output B of the TB is given in Tab. II, its formal description is $\bar{B} = T \wedge B$.

TABLE II: Truth Table of the T-Buffer

\bar{B}	T	\bar{B}	Comment
-	0	0	Reset
0	1	1	Oscillation
1	1	0	Oscillation

Fig. 6a shows the schematic of the TB in TL. For the purpose of a consistent and structural analysis for delay-insensitive circuits, the symbol of the TB has been chosen as presented in Fig. 6b.

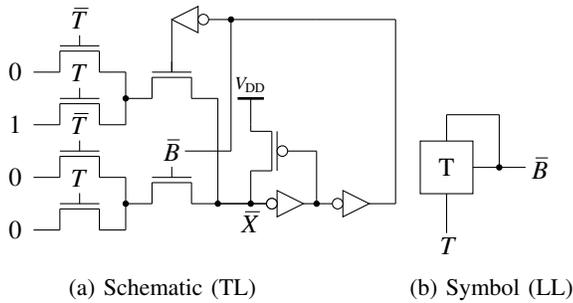


Fig. 6: T-Buffer in Transistor Level and Logic Level

B. Measurement circuit

According to the time delay estimates of Vivado, the FLPD of the TB is expected to be in the hundreds of picoseconds range. This results in frequencies of several gigahertz for the oscillation of the TB. Therefore, in order to measure this delay of the TB, regardless of the specifications of the available oscilloscope, an asynchronous frequency divider is required

to downscale the frequency of the oscillation prior to measurement. The N-stage asynchronous frequency divider used is shown in Fig. 7. It consists of a LUT-based data latch (D-Latch) and an N-1 stage asynchronous counter. The D-Latch

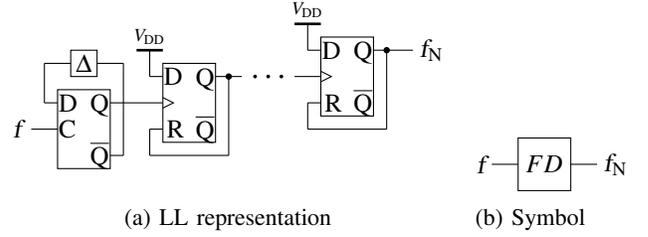


Fig. 7: N-stage, asynchronous frequency divider

works as a single-stage frequency divider. It has proven to be necessary, as the subsequent counter would otherwise not be able to detect every single positive edge, especially for oscillation of the TB with relatively high frequencies. However, for the LUT-based D-Latch to work as an asynchronous frequency divider, the data signal must correspond to the current state of the latch but switched and the FLPD of the latch must be the same as the FLPD of the TB. To guarantee the equality of the delays, the same LUTs of different CLBs are used and constrained in the same way, resulting in identical routing of the wires inside the FPGA and equal delay estimates of Vivado [2]. Furthermore, it was proven that the measurement result for an oscillation of the TB with a relatively low frequency does not change when replacing the D-Latch with another D-FF-based counter stage. The asynchronous counter is realised as a ripple-through counter [9] using the available D-FF in the FPGA. By setting the data input D of the FF to a logic 1 and using the synchronous reset to generate the logic 0, the additional inverter of the classical structure with D-FF is saved leading to a much higher maximum switching frequency of the counter [14].

C. Measurement

The TB together with a 10-stage frequency divider was implemented in a Xilinx Artix-35T FPGA (xc7a35ticsg324-1L), which contains 5200 slices with four 6-input LUTs and eight FFs each [1]. The FPGA was programmed using Vivado 2020.2 and the source codes were written in Verilog. Our sample consisted of 16 LUTs allocated in the four slices X0Y0, X1Y0, X2Y0 and X3Y0 of the FPGA. For each input of a LUT ten measurements were performed resulting in a total of 960 measurements. Throughout the experiment, the die temperature was maintained at $(31.7 \pm 0.6)^\circ\text{C}$. The signal was recorded using the Tektronix TDS1012B oscilloscope with a record length of 2500pts and a sample rate of up to 1GS/s for periodic signals. Considering that the 10-stage frequency divider scales down the oscillating signal by a factor of $2^{-10} \approx 10^{-3}$ to several megahertz, a time interval of $5\mu\text{s}$ in combination with a sample rate of 500MS/s was found as an appropriate trade-off between number of bilevel pulses per record and the resolution of a single pulse.

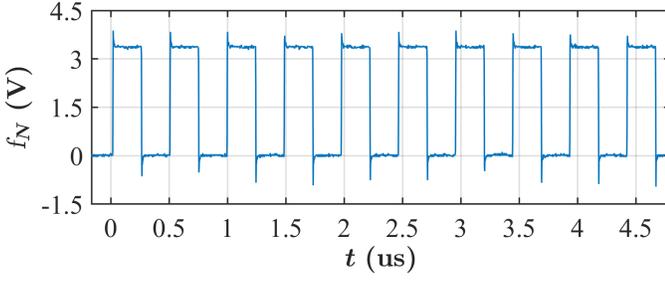


Fig. 8: Oscillating signal for a TB in LUT A of Slice X0Y0 using input A5 as its feedback path

Fig. 8 shows the oscillating signal measured for a TB implemented in LUT A of Slice X0Y0 using input A5 for the feedback signal. To calculate the FLPD, the average of the widths of all bi-level pulses measured over ten instances was first determined and then scaled down by a factor of 2^{-10} to compensate for the 10-step frequency divider. For completeness, the corrected sample standard deviation was also calculated. The generalized formulas are given in Eq. 4 and Eq. 5 [11].

$$\tau_{FLPD,avg} = \frac{1}{M(T-1)} \sum_{i=1}^M \sum_{j=1}^{T-1} \frac{t_{i,j+1} - t_{i,j}}{2^N} \quad (4)$$

$$\sigma_{FLPD}^2 = \frac{1}{M(T-1)-1} \sum_{i=1}^M \sum_{j=1}^{T-1} \left(\frac{t_{i,j+1} - t_{i,j}}{2^N} - \tau_{FLPD,avg} \right)^2 \quad (5)$$

Parameter N contains the number of stages of the frequency divider, M represents the number of measurements taken and parameter T marks the number of transitions of the recorded signal at 1.65V as the mid voltage level between a logical 0 and a logical 1 in the case of the Artix-35T FPGA. The time variables $t_{i,j+1}$ and $t_{i,j}$ carry the linearly interpolated time values for two successive transitions at 1.65V of the i -th measured signal. The results of calculating the mean value and standard deviation of the FLPD for the signal given in Fig. 8 and the signals obtained from using the other inputs of the LUT for the feedback signal, are listed in Tab. III.

TABLE III: Mean FLPD of the TB in LUT A of slice X0Y0

	A1	A2	A3	A4	A5	A6	
$\tau_{FLPD,avg}$	595.1	582.2	477.9	435.8	239.2	290.3	[ps]
σ_{FLPD}	0.6	0.6	0.5	0.6	0.5	0.5	[ps]

Evidently, the FLPD is minimal when using input A5 for the feedback signal and not input A6 as the structure of the TB in Fig. 5 would suggest. Reason is the difference in routing of the two implementations, which the comparison of Fig. 9a and Fig. 9b reveals. Fig. 10 shows the average FLPD results for all 16 LUTs. A distinction was only made between the LUTs, but not between the slices, as similar results were expected for different slices. The standard deviation is not shown because it is too small to be displayed on the y-axis. The width of

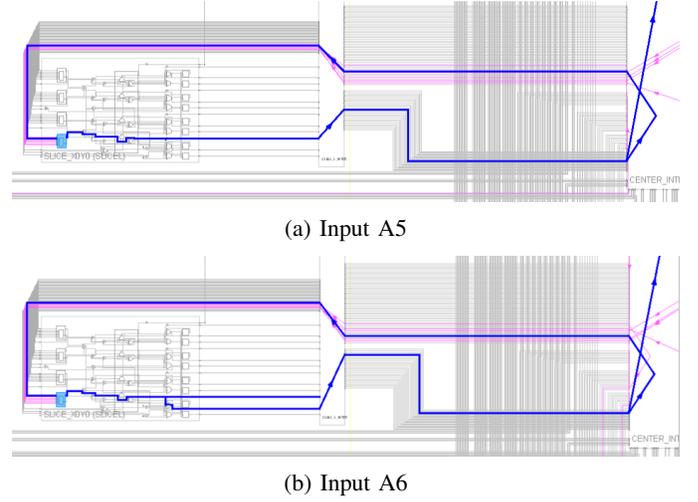


Fig. 9: Implementation of the TB in LUT A of Slice X0Y0 using input A5 or A6 for its feedback path (highlighted)

the horizontal lines in the figure has no significance and is only used for readability. The minimum FLPD measured was 220.7ps with a standard deviation of 0.6ps for LUT D in Slice X1Y0 using Input A5. The maximum FLPD measured was 759.6ps with a standard deviation of 0.6ps for LUT B in Slice X1Y0 using Input A2. Regarding the minimum FLPD,

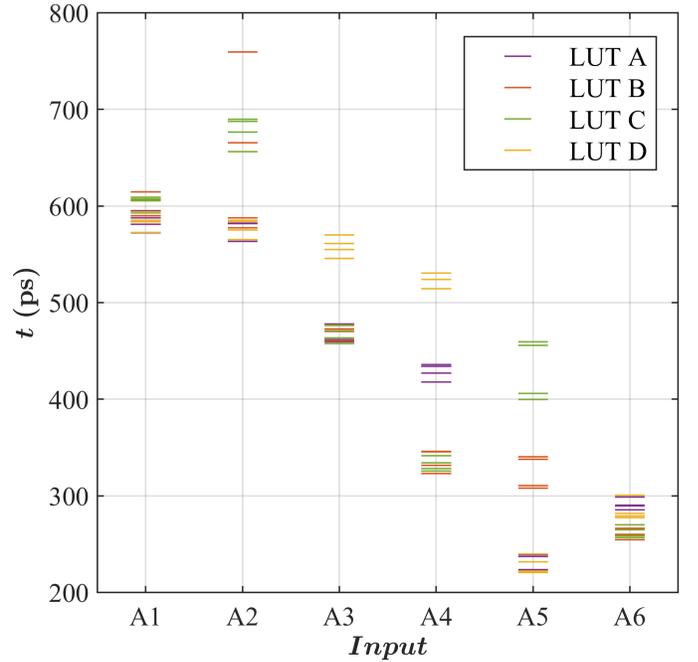


Fig. 10: FLPDs for the TB implemented in LUT A-D of Slice X0Y0-X3Y0

the results are only consistent with Tab. III for the LUTs A and D but not for the LUTs B and C. Again, the difference in routing is the cause. For completion, Tab. IV lists the minimum FLPD for each of the 16 LUTs. For further consideration of the Muller C, LUT A of slice X0Y0 is used, with the feedback

Slice	$\tau_{FLPD,avg}$	σ_{FLPD}	Slice	$\tau_{FLPD,avg}$	σ_{FLPD}
X0Y0	239.2	0.5	X0Y0	260.3	0.6
X1Y0	221.0	0.5	X1Y0	254.6	0.6
X2Y0	237.2	0.5	X2Y0	266.3	0.6
X3Y0	223.5	0.5	X3Y0	254.6	0.6

(a) Input A5, LUT A

(b) Input A6, LUT B

Slice	$\tau_{FLPD,avg}$	σ_{FLPD}	Slice	$\tau_{FLPD,avg}$	σ_{FLPD}
X0Y0	264.8	0.5	X0Y0	231.7	0.5
X1Y0	256.7	0.6	X1Y0	220.7	0.6
X2Y0	270.0	0.6	X2Y0	239.9	0.5
X3Y0	258.6	0.5	X3Y0	222.2	0.9

(c) Input A6, LUT C

(d) Input A5, LUT D

TABLE IV: Minimum FLPD for the LUTs A-D of the Slices X0Y0-X3Y0

line connected to input A5, as this configuration results in a minimum FLPD. The exact structure is shown in Fig. 11 with the FLPD being the sum of the forward path delay τ_δ and the reverse path delay τ_Δ .

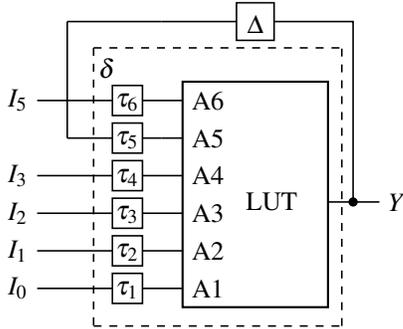


Fig. 11: Feedback mapped to a six input LUT

VI. TEST

To ensure that the relation $\tau_\Delta < \tau_\delta$ is valid, a fault model is set up, which represents the behavior in case of a fault ($\tau_\Delta > \tau_\delta$) and is tested for this fault. For the test, a pulse circuit is designed, which generates a short pulse of length $\tau \geq \tau_\delta$ when input i is switched. If $\tau_\Delta > \tau_\delta$, it must be guaranteed that the relation $\tau \leq \tau_\Delta$ holds, otherwise the error cannot be detected at the output. The pulse then is the input for a feedback OR circuit, see Fig. 12. The fault model states that if the OR

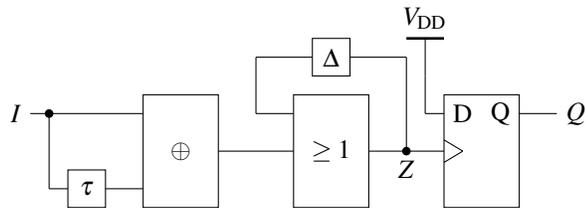


Fig. 12: Test Circuit for Verification

circuit does not set to 1, but has passed a pulse, the error

and the relation $\tau_\Delta > \tau_\delta$ is valid. To ensure that a pulse has actually been passed, the Clk input of a D-FF is connected to the output of the OR circuit. The D-input is set to V_{DD} , so that if a pulse occurs, the D-FF stores a 1. So in case of an error, after a certain time t_1 there is a 0 at the OR and a 1 at the D-FF. If the assignment [11] is visible at the outputs (z, q), a function stable Muller C-element has been implemented. The expected signals for the fault model can be seen in Fig. 13, where $\tau = \tau_\delta$ holds.

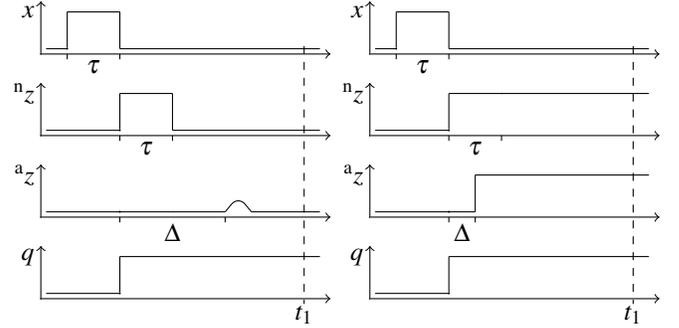
(a) Fault Case $\tau < \Delta$ (b) Fault Free Case $\tau > \Delta$

Fig. 13: Test Cases

The test now reads:

$$\begin{aligned} n_z(t_1) = 0 & \quad \text{fail} \\ n_z(t_1) \neq 0 & \quad \text{pass} \end{aligned}$$

and says that the error condition exists if the output is 0. Thus, a 0 at the output then clearly indicates that the reverse path is slower than the forward path and there is no function stable design. The results of the test show that if data input A6 is used as data input, the error occurs and the Muller C element is not functionally stable. On the other hand, if data inputs A4 to A1 are used, the error no longer occurs and it is logically concluded that the circuit is function stable. Therefore, for the fastest and safest Muller C element, the faster data inputs A4 and A3 are preferred, see Fig 14, where the unused input pins are connected to V_{DD} . This function stable Muller C can now

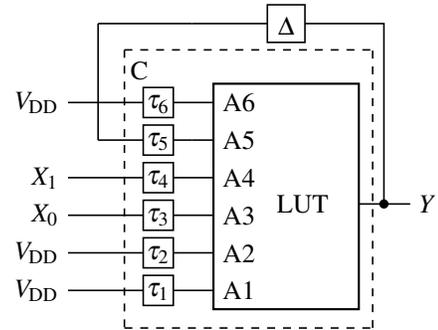


Fig. 14: Function Stable Muller C-element mapped to a six input LUT

be used to implement safe asynchronous circuits in FPGAs.

VII. CONCLUSION AND FUTURE WORK

This paper first summarized the types of stabilization of a digital circuit and then showed how a feedback LUT can realize a function stable Muller C-element. It is important that the forward delay is larger than the feedback delay in order to stabilize a transition or not to change the value at all in case of an error when the pulse is too small. With the measurements shown in this paper, the LUTs can be used to create function stable circuits. In the case discussed here, LUT A was used in slice X0Y0 and it was shown that the Muller C-element is function stable when the data input is placed on A4 and the feedback input is placed on A5. This knowledge can be used to implement asynchronous circuits in the FPGA. For example, the Muller C element can now be used to implement function stable Muller C pipelines. In the future, function stabilization will be used to implement various asynchronous safe basic structures such as latches and flip-flops. Furthermore, the main goal is to realize dual-rail domino logic circuits that are self-locking (self-X), self-timed, hazard and race-free, and thus globally asynchronous, locally synchronous and maximally safe.

REFERENCES

- [1] *7 Series FPGAs Configurable Logic Block*. XILINX, UG474 (v1.8) September 27, 2016.
- [2] *Vivado Design Suite User Guide*. XILINX, UG903 (v2020.1) August 17, 2020.
- [3] *Vivado Design Suite 7 Series FPGA Libraries Guide*. XILINX, UG953 (v 2012.2) July 25, 2012.
- [4] Peter Brungs and Marcel Baunach. Einsatz von dynamisch rekonfigurierbaren fpgas in fahrzeugen. In Heinrich C. Mayr and Martin Pinzger, editors, *Informatik 2016*, pages 1565–1577, Bonn, 2016. Gesellschaft für Informatik e.V.
- [5] Charles Chiasson and Vaughn Betz. Should fpgas abandon the pass-gate? In *2013 23rd International Conference on Field programmable Logic and Applications*, pages 1–8, 2013.
- [6] Sarah Harris and David Harris. *Digital Design and Computer Architecture: ARM Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2015.
- [7] K. Maheswaran. *Implementing Self-timed Circuits in Field Programmable Gate Arrays*. University of California, Davis, 1995.
- [8] Matheus Moreira, Bruno Oliveira, Fernando Moraes, and Ney Calazans. Impact of c-elements in asynchronous circuits. In *Thirteenth International Symposium on Quality Electronic Design (ISQED)*, pages 437–343, 2012.
- [9] Noel M. Morris. *Asynchronous Counters*, pages 105–109. Macmillan Education UK, London, 1974.
- [10] Tertulien Ndjountche. *Digital electronics 3 : finite-state machines*. Electronics engineering series (London, England). ISTE, Ltd ; Wiley, London, UK : Hoboken, New Jersey, 2016.
- [11] R Parthier. *Messtechnik: Vom SI-Einheitensystem über Bewertung von Messergebnissen zu Anwendungen der elektrischen Messtechnik*. Springer Vieweg, 2020.
- [12] Cuong Pham-Quoc and Anh-Vu Dinh-Duc. Hazard-free muller gates for implementing asynchronous circuits on xilinx fpga. In *2010 Fifth IEEE International Symposium on Electronic Design, Test & Applications*, pages 289–292, 2010.
- [13] P. Srivastava. *Completion Detection in Asynchronous Circuits: Toward Solution of Clock-Related Design Challenges*. Springer International Publishing, 2022.
- [14] Shashank Uniyal and Vishal Ramola. A new 4 bit asynchronous counter using novel low power explicit type pulse-triggered delay flip flop (d-ff) 1, 01 2019.
- [15] Gürkan Uygur and Sebastian M. Sattler. A real-world model of partially defined logic. In *12th International workshop on Boolean Problems*, 2016.
- [16] Heinz-Dietrich Wuttke and Karsten Henke. *Schaltsysteme*. Informatik. Pearson Studium, München, 2003.
- [17] Hans Joachim Zander. *Logischer Entwurf binaerer Systeme*. VEB Verlag Technik, 1989.