Measurements of metastability in MUTEX on an FPGA

Nguyen Van Toan, Dam Minh Tung, and Jeong-Gun Lee

Abstract: In this paper, we propose a new method for measuring metastability in the mutual exclusion element (MUTEX) implemented on a Field Programmable Gate Array (FPGA). Our method uses fine-grained phase shifts of a digital clock manager to trigger Flip-Flops to generate concurrent inputs for a MUTEX. By dynamically adjusting the phase shift between two clock signals, we can force the MUTEX into a metastable state. The benefit of our approach is that it is easier to force the MUTEX become metastable compared to the conventional approach using two un-correlated signals. The experiments have been performed on a Xilinx Spartan-6 (XC6SLX9-4TQG144C).

Keywords: metastability, globally asynchronous locally synchronous (GALS), mutual exclusion element (MUTEX), field programmable gate array (FPGA), stoppable/stretchable clocking

Classification: Integrated circuits

References

[1] R. Dobkin, et al.: “High rate data synchronization in GALS SoCs,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 14 (2006) 1063 (DOI: 10.1109/TVLSI.2006.884148).
[2] Zh. Yu and B. M. Baas: “High performance, energy efficiency, and scalability with GALS chip multiprocessors,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 17 (2009) 66 (DOI: 10.1109/TVLSI.2008.2001947).
[3] X. Fan, et al.: “GALS design for on-chip ground bounce suppression,” 17th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) (2011) 43 (DOI: 10.1109/ASYNC.2011.11).
[4] X. Fan, et al.: “GALS design for spectral peak attenuation of switching current,” 19th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) (2013) 83 (DOI: 10.1109/ASYNC.2013.28).
[5] B. Keller, et al.: “A pausable bisynchronous FIFO for GALS systems,” 21st IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) (2015) 1 (DOI: 10.1109/ASYNC.2015.9).
[6] X. Fan, et al.: “Performance analysis of GALS datalink based on pausable clocking,” 18th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) (2012) 126 (DOI: 10.1109/ASYNC.2012.24).
[7] T. Polzer and A. Steininger: “An approach for efficient metastability
characterization of FPGAs through the designer,” 19th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC) (2013) 174 (DOI: 10.1109/ASYNC.2013.14).

[8] Th. Polzer, et al.: “A programmable delay line for metastability characterization in FPGAs,” 24th Austrian Workshop on Microelectronic (2016) 51 (DOI: 10.1109/Austrochip.2016.021).

[9] S. Beer, et al.: “Metastability challenges for 65 nm and beyond; simulation and measurements,” Design, Automation & Test in Europe Conference & Exhibition (DATE) (2013) (DOI: 10.7873/DATEx.2013.268).

[10] D. J. Kinniment, et al.: “Measuring deep metastability and its effect on synchronizer performance,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 15 (2007) 1028 (DOI: 10.1109/TVLSI.2007.902207).

[11] T. Polzer, et al.: “On the appropriate handling of metastable voltages in FPGAs,” J. Circuits Syst. Comput. 25 (2016) 1640020 (DOI: 10.1142/S021812661640020X).

[12] D. J. Kinniment: in Synchronization and Arbitration in Digital Systems (Wiley, West Sussex, 2007) 23.

[13] T. Polzer and A. Steininger: “Metastability characterization for Muller C-elements,” 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS) (2013) 164 (DOI: 10.1109/PATMOS.2013.6662170).

[14] Xilinx Inc.: Spartan-6 FPGA Clocking Resources (2015) 57.

1 Introduction

With rapid semiconductor technology scaling, the number of transistors integrated on a single chip die has been increased exponentially. The high integration densities allow system on chips (SoC) become bigger and more complex circuits. However, the modern SoCs built in deep process technology nodes face some challenges. The increase of wire delays, process, temperature, voltage (PVT) variations make timing closures become difficult issues that take much time and efforts of SoCs’ designers in design and verification [1, 2]. Particularly, it is more difficult to distribute a huge global clock network over entire chip with low clock skews. Additionally, power consumption of a huge clock network is also another problem that designers must take care of to meet the power/energy consumption requirements.

As an alternative approach, multiple clock domains can be exploited for a chip. When each clock domain becomes smaller, it is easier to achieve the timing closure, and maintain low clock skews. Furthermore, the power/energy consumption can be saved when each locally synchronous module works with its optimal clock frequency [2]. Since different modules use unrelated clock frequencies, such a system can be called a globally asynchronous locally synchronous (GALS) system [1]. Another benefit of the GALS system is the electromagnetic inference (EMI) can be reduced since their locally synchronous modules operate either at different frequencies or at different phases [3, 4]. Hence, the switching activities inside a GALS system can be spread over time. However, the inter-communications between synchronous modules are still challenges. When two or more parties communicate with each other asynchronously, synchronization mechanisms for them must be designed carefully since the metastability can be occurred. A simple
“brute-force” synchronizer (two or more cascaded FFs) suppresses the probability of metastability to a negligible value [5]. Unfortunately, it introduces a high latency that degrades the performance of a system. A dual-clock first-inputs-first-outputs (FIFO) can be also a preferable choice for hardware designers. However, latency and area overheads are problem for this approach. Additionally, for dual clock FIFO, there still exists the metastability with a negligible value. A pausable/stretchable clocking based GALS system is a promising approach that can solve high latency communications and eliminate the metastability in data-paths.

In GALS systems using a pausable clocking scheme, their locally synchronous components communicate with each other by using a handshake protocol. The locally synchronous modules are wrapped around by port controllers [1, 6]. These port controllers are in charge of handling the handshake signals. They also communicate with the pausable clock generators to temporarily pause or release the local clock signal. When locally synchronous modules have needs to transfer data, they send requests to port controllers. Then, port controllers will request to stop the clock generators temporarily for safely sampling the asynchronous data.

However, there is another issue which is related to the arbitration between requests from the port controller and the local clock signal. Normally, a “MUTEX” is utilized to arbitrate possible conflicts between the requests and the local clock. When these signals arrive at the MUTEX at the same time or within a very narrow time window, the MUTEX can enter the metastable state. In this case, it needs more time to judge which signal wins over the other. There is an unknown resolution time for a MUTEX. A long resolution time will adversely impacts on the total throughput of a GALS system. In the design phase, hardware designers need to verify their MUTEX designs carefully.

Two approach for characterizing the metastability in state elements: random approach and deterministic approach [7]. In the random approach, input signals (data and clock) for state elements are independent. The probability which inputs overlap (i.e., rising edges overlap), or be close in time is uniformly distributed. In deterministic approach, the inputs are concentrated on an interest region controlably. In [7], and [8], they applied the random approach to characterize the metastability for a Flip-Flop on an FPGA Virtex-4. Similarly, the work in [9] characterized the metastability of Flip-Flips on 65 nm CMOS technology. With the random approach, there is an extremely small number of input stimuli that can cause very long metastable responses. This means it takes a very long measurement time to obtain a reliable Mean Time Between Failures (MTBF) [10]. Thus, authors in [10] proposed a deterministic approach to characterize the metastability of a Flip-Flop. Their approach used an Operational Amplifier and some discrete components to control the arrival times of data and clock signals. However, their approach is inevitable in an FPGA.

In this paper, we propose a method for measuring the metastability resolution time of a MUTEX implemented in an FPGA by employing the dynamically configurable phase shift of the DCMs that are available on a Xilinx FPGA. The MUTEX under test is built by cross-coupling two NAND gates in an FPGA manually. The controllable phase shift feature of the DCM allows us to generate concurrent request signals to the inputs of the MUTEX. Therefore, we can easily
force the MUTEX into a deep metastable state to measure the resolution time of the metastability in the worst case. This cannot be implemented easily with the traditional method which uses two individually uncorrelated signals for measuring the metastability resolution time.

The paper is organized as follows. Section 2 presents the preliminary with the overview of a MUTEX and metastability. The designs of a metastability detection circuit and a clock phase adjustment controller are proposed and presented in Section 3. The measurement procedure and experimental results are presented in Section 4. Finally, Section 5 summaries our paper.

2 Preliminaries

This section describes the basic operation of a MUTEX, the metastability phenomenon in the MUTEX, and the role of the MUTEX in a pausable/stretchable clocking based application.

2.1 Mutual exclusion element (MUTEX) and metastability

A conventional MUTEX is made by two cross-coupled NAND gates. Then, their outputs are inverted to obtain the grant signals as shown in Fig. 1 [11]. In full-custom design, the two cross-coupled NAND gates can be optimized so that the time delays of feedback paths are as minimal as possible. Eventually, designing a MUTEX is very similar to design a set-reset (SR) latch. In an FPGA, two cross-coupled NAND gates are not available. In this work, each NAND gate can be manually implemented by using a look-up table (LUT). These LUT-based NAND gates should be placed as closely as possible to help MUTEX resolve metastability quickly if any [12]. The MUTEX works with the basis of first-come-first-served. Two asynchronous signals can arrive at the input of a MUTEX at the same time or in a very narrow time window. In this case, the MUTEX outputs can have metastable voltages, which means that the MUTEX needs more time to determine which request will be served first. Normally, a filter can be attached to the MUTEX outputs to avoid the metastable signals propagating to the successive circuits. Two inverters with low voltage thresholds are proven as good circuits to filter out the metastability [11]. These filters will keep their outputs unchanged (logic-0) until the MUTEX exactly determines which request wins. The metastable signals sometimes make the successive circuits malfunctions or reduce the performance of the system. In this work, the circuit structure of the MUTEX is used to measure the metastability as shown in Fig. 1. The output voltage of a MUTEX is illustrated in Fig. 2.
In the case that two requests arrive at the MUTEX at the same time or very close in time, the output voltages of the MUTEX can be unknown, typically VDD/2.

### 2.2 MUTEX in a pausible clocking based GALS system

In GALS systems based on a pausible clocking scheme, a MUTEX is used to arbitrate two or more input signals: requests and the local clock signal. The pausible clock based GALS system is illustrated in Fig. 3.

If the request Ri arrives at the MUTEX first, then it will be served by the MUTEX, and the acknowledge signal Ai will be granted. If the local clock signal “clk_r” arrives at the MUTEX first, the output signal “grt” will be granted, and the ring oscillator continues its operation. In this case, the request Ri must wait until the clk_r signal finishes its HIGH duration, which normally equal to a half of clock cycle. In case of two signals Ri and “clk_r” being overlapped or very close in time, the MUTEX can enter the metastable state at which its output voltage levels before the filter are in the unknown region. Thanks to the metastability filter, the successive circuits still work properly. If the metastability lasts so long, or the request Ri wins and lasts so long (longer than a half of clock cycle), the next rising edge of the local clock will be stretched. The Muller-C element in Fig. 3 is used to synchronize “grt” and “clk_r” signals, and to make sure the minimum pulse width of the clock signal. The basic operation of the Muller-C is described in [13].
3 Circuit design

In this section, firstly we describe the design and the operation principle of the metastability detection circuit in details. Then, a finite state machine that is used to control the clock phase adjustment for measuring the metastability resolution time is also mentioned.

3.1 Metastability detection design

The circuit schematic for the metastability detection is shown in Fig. 4. The first digital clock manager (DCM), DCM-1, is responsible for multiplying/dividing the input clock frequency. In our case, the external clock frequency is 50 MHz. The clock frequency for the metastability detection circuit is 25 MHz. The output clock signal of DCM-1 is the clock source for the DCM-2, DCM-3, and the request generation circuit (one D-FF and a feedback inverter). The output clock signal of the DCM-2 is phase-shifted with $\alpha$ degrees compared to its input clock signal. The phase shift of $\alpha$ can be dynamically changed by configuring its input values. By changing the phase shift $\alpha$ of the DCM-2, we can align two input request signals, R1 and R2 of the MUTEX, that are generated by two D-FFs.

In the circuit, the input signal "Calib" is used to determine the nominal propagation delay of the MUTEX including its input and output wire delays. By setting the signal "Calib" to logic-0, the request signal R2 is always at logic-0. The request signal R1 is always served by the MUTEX. By changing the phase shift $\alpha$ of the DCM-2, we can align two input request signals, R1 and R2 of the MUTEX, that are generated by two D-FFs.

At the rising edge of the clock signal (at $\delta$ pin) of the DCM-3, the detection FF captures the grant signal G1. We assume that the metastability of the MUTEX can be resolved within a half of clock cycle of the DCM-3 ($\approx 20$ ns). The reference FF captures the grant signal G1 by using the DCM-3 clock signal with 270° (This is equivalent to 30 ns) later than the clock signal $\delta$. The grant signal is captured after this amount of phase/delay for sampling the fully stabilized version of the signal G1 without metastability. The synchronization FF is used to synchronize the data of detection FF and the reference FF. If these two values are different from each other,
there is a change in the grant signal $G_1$ between the rising edges of the clock signal $\delta$ and the clock signal $270^\circ$. The timing diagram of the metastability detection circuit is shown in Fig. 5.

The schematic of the pulse extension is illustrated in Fig. 6. The pulse extension circuit helps to increase the HIGH duration of the signal that connects to the request $R_2$. The purpose of this circuit is to support the phase adjustment between two requests. If we change the phase $\alpha$ until there is no assertion for $G_1$, we will know that $R_2$ completely wins over the request $R_1$ since the HIGH duration of $R_2$ completely covers the HIGH duration of $R_1$. At that time, we will change the phase $\alpha$ in an opposite direction to find the point where the probability of overlapping of two input signals is highest.

3.2 Clock phase adjustment controller

The finite state machine (FSM) for dynamically adjusting the phase shift of a DCM is depicted in Fig. 7. In this FSM, there are 5 states: IDLE, INCR, INCR_PHASE_STEP, DECR, and DECR_PHASE_STEP. Initially, the machine is at IDLE, and it still maintains this state if the enable signal, $psen$, is still inactive. When the signal $psen$ is active HIGH, and the current phase ($cur\_ps$) is not equal to the target phase ($targ\_ps$), it passes to the INCR (increase the phase) or DECR (decrease the phase) state depending on the value of $psincdec$. The DCM will increases the phase by one
degree if \( psincdec \) equals 1 or decreases the phase by one degree if \( psincdec \) equals 0. Then, when the phase shifting is done, the signal \( psdone \) is active HIGH for one clock cycle. The state machine moves to the state INCR\_PHASE\_STEP or DECR\_PHASE\_STEP if its previous state is INCR or DECR. At these states, if the current phase shift is still not equal to the target phase shift (\( cur\_ps \neq targ\_ps \)), the machine will move back to the INCR or DECR. Otherwise, it moves to IDLE state. In this FSM, the reset signal is not shown. Whenever the reset signal is asserted, the FSM will move to the IDLE state. For Spartan-6 FPGA, the phase shift is increased or decreased by 25 ps for each step. For more details of the DCM specification, we can refer [14].

![Fig. 7. The FSM for dynamically adjusting the phase shift of a DCM.](image)

---

4 Metastability measurement and results

4.1 The procedure of metastability measurement

It is difficult to force the MUTEX enter the metastable state if we use two uncorrelated signals to feed its two inputs since the probability of the conflict between these two signals is very low, and the natural unbalance of the MUTEX circuit can help resolve the conflict. Those reasons can lead to the long measurement time. In order to make measurements become easier, we connect two phase-related signals to its two inputs. These two phase-related signals are generated by two Flip-Flops that are clocked by two phase-related clock signals which are shown in Fig. 4. The measurement procedure is described as follows:

- At first, the signal “Calib” is externally tied to logic-0 to de-assert the input request R2 of the MUTEX.
- Adjust (increase/decrease) the phase shift of the output clock signal of DCM-3 (\( \delta \) pin) so that there are errors observed at the output of Error-FF.
- Once again, adjust (increase/decrease) the phase shift of the output clock signal of DCM-3 (\( \delta \) pin) until those errors disappear. The latest phase shift of the DCM-3 determines the delay from the input to the output of the MUTEX. That phase shift (\( PS_{nominal} \)) is equivalent to the nominal propagation delay of the MUTEX (including input and output wire delays). When the request R2 is allowed to pass the input R2 of the MUTEX, and if the R2 and R1 cause the metastability in the MUTEX, so the propagation delay of the MUTEX can be increased. In this case, we have to increase the phase shift \( \delta \) to extend the
capturing time so that we can capture correct data without metastability induced errors. That phase shift $\delta$ consists of the nominal propagation delay of the MUTEX (PSnonimal) and the metastability resolution time. Finally, the metastability resolution time is calculated by subtracting the PSnonimal from the latest phase shift $\delta$.

- Now, connect the signal “Calib” to logic-1 to assert the input request $R2$ of the MUTEX. At this moment, in order to make two input requests $R1$ and $R2$ become conflict (i.e., both $R1$ and $R2$ arrive at the MUTEX at the same time), we dynamically adjust the phase shift $\alpha$ of the DCM-2 until there are errors observed at the output of Error-FF.
- By gradually increasing the phase shift of the DCM-3, we can measure the time resolution of the MUTEX on an FPGA.

### 4.2 Experimental results

In the experiments, the external clock frequency is 50 MHz while the output clock frequency of all DCMs are configured to generate the clock frequency of 25 MHz. The experimental results are summarized in Fig. 8 and Fig. 9.

We start from the case of the highest metastability rate. To do so, first we calibrate the system to determine the nominal propagation delay of the MUTEX. The calibration procedure is described in Section 4.1. The phase shift $\delta$ of the DCM-3 at which there is no error monitored at the Error-FF determines the nominal propagation delay of the MUTEX. Then, we gradually change the phase shift $\alpha$ of the DCM-2 in positive or negative direction so that the requests $R1$ and $R2$ overlap or are very close to each other with the support of the pulse extension circuit as shown in Section 3.1. The purpose of this step is to force the MUTEX enter a deep metastability to obtain the highest metastability rate. The procedure is continued by gradually increasing the phase shift $\delta$ in positive direction so that the MUTEX has more time to resolve the metastability. By doing this, the number of error monitored at the Error-FF will be decreased. The total of failure events in this measurement is counted for one million (1,000,000) events which the request $R1$ wins over the request $R2$. There are two counters to count the total events and failure events, but they are not shown in Fig. 4.

In the experiments, the phase shift $\delta$ is changed by 125 ps after each measurement. The phase shift resolution of the DCM on Spartan-6 can be achieved up to 25 ps. To reduce the measurement time, we decide to increase the phase shift by 125 ps for each step. As can be seen in Fig. 8, when increasing the phase shift $\delta$ (positive direction), the number of failure events are decreased rapidly. That means the number of failure events are decreased when the MUTEX has more time to resolve the metastability. The failure rate is rapidly decreased from 0 to 2 ns. However, it is decreased very slowly from 2 ns to 6 ns. After 6 ns, there is no observed failure event which means that all the metastability is resolved. Fig. 9 shows the details of the failure events from 4 ns to 6.5 ns of Fig. 8. With 4 ns resolution time, 16 failure events happen and then the number of failure events becomes two with further 1 ns resolution time.

Lastly, we compare the quality of the metastability behavior in the manual design MUTEX on an FPGA with a Flip-Flop (DFF) that is available as a well-
made primitive in an FPGA. For the comparison, we also apply the proposed metastability measurement method to measure the metastability resolution time of the DFF. Even though the functions of the DFF and the MUTEX are different to each other (a DFF is used to capture an input data at a clock event while the MUTEX is used to decide which event happens first among multiple events), we can compare the metastability resolution times of those two elements in order to understand the relative quality of the MUTEX. The metastability measurement results of the DFF are summarized in Fig. 10.

Fig. 8. The metastability measurement results of a MUTEX.

Fig. 9. The more details of the metastability measurement results of a MUTEX from 4 ns to 6 ns.

Fig. 10. The metastability measurement results of a Flip-Flop.
The more detailed measurement results of the DFF at the resolution time from 350 ps to 750 ps are illustrated in Fig. 11. The metastability resolution time of the DFF for having zero failure in this experiment is about 725 ps and the time is much shorter than that of the MUTEX (about 6.25 ns). It shows the limit of the metastability resolution time in the MUTEX that has to be implemented in a manual way due to the lack of a MUTEX primitive in a modern FPGA architecture.

5 Conclusion

In this paper, we have proposed and implemented the metastability detection circuit to measure the metastability resolution time of a MUTEX implemented on an FPGA. Our approach is based on the dynamically configurable phase shift of the DCM that is available in XilinxFPGAs. The benefit of our method is that we can force the MUTEX to a deep metastable state easily. The experimental results show that the failure events of the MUTEX are rapidly decreased with the increase of the resolution time at the initial stage. However, the metastability resolution rate is rapidly decreased, and it takes more time to resolve all failure events as described in Fig. 8 and Fig. 9.

Acknowledgments

This work has been supported by Basic Science Research Program through the National Research Foundation (2015R1D1A3A01019869). The work has been also partially supported by the Leading Human Resource Training Program of Regional Neo-Industry through the National Research Foundation (2016H1D5A1910630).