Design and Analysis of Testable Mutual Exclusion Elements

Yang Zhang*, Leandro S. Heck†, Matheus T. Moreira*†, David Zar‡,
Mel Breuer*, Ney L. V. Calazans†, and Peter A. Beerel*§
*University of Southern California - Los Angeles, United States
†Pontifícia Universidade Católica do Rio Grande do Sul - Porto Alegre, Brazil
‡Blendics - Saint Louis, United States
{zhan808, pabeerel}@usc.edu, {matheus.moreira, leandro.heck}@acad.pucrs.br, dzar@blendics.com, ney.calazans@pucrs.br

***Abstract*—Mutual exclusion elements (MUTEXes) are fundamental components of asynchronous arbiters and are particularly critical to ensure metastable signals are properly filtered before reaching the arbiter outputs. However, despite their importance, the testability of these circuits is typically limited to functional testing. This paper discusses why this is not sufficient and addresses testability issues in both full-custom and standard-cell implementations. In particular, it proposes two new testable implementations that not only ensure improved coverage for single stuck-at faults but also enable testing the filtering of metastable signals. Additionally, this article quantifies the cost of the testable designs by comparing them to similar traditional designs in terms of area, power and metastability resolution time. Results show the proposed optimizations do increase area and power but have small impact on performance.**

***Index Terms*—Mutual exclusion element, MUTEX, arbiter, testability, metastability, τ, tau, asynchronous circuits.**

I. Introduction

The discrete notion of time in synchronous systems makes arbitration between competing requests to a shared resource a relatively simple task, relying on simple clocked finite state machines. If two requests arrive during the same clock cycle, one is given priority as defined by a given arbitration policy (round robin, strict priority, etc.) implemented using standard combinational logic. Because of their continuous time nature, arbitration in an asynchronous circuits works quite differently, generally on a first-come-first-served basis. Determining who arrives first in the continuous time domain can be challenging when requests are close in time and requires specialized components called mutual exclusion elements (MUTEXes). In fact, resolving which request arrives first is so difficult that the time for any MUTEX implementation to make this decision is unbounded [1] [2] [3]. In particular, decisions rely on MUTEX internal non-digital behavior to determine which request to grant first. Moreover, before the decision, one or more internal signals can remain in a metastable state. For this reason, a MUTEX includes specialized *metastability filters* (MFs) that ensure outputs do not switch until the components make a decision. These filters are expected to guarantee that metastable states do not propagate to downstream logic, which can lead to unpredictable and irreversible effects. This filter is expected to guarantee that metastable states do not propagate to downstream logic, which can lead to unpredictable and irreversible effects.

MUTEXes are useful in a variety of asynchronous designs including register files [4], crossbars, clock domain converters [5], and NoC routers [6]–[8]. However, despite the importance of MUTEX components in asynchronous design, research literature has overlooked the problem of testing such components. In fact, most works rely on functional test approach only [9]. The specialized nature of MUTEXes makes testing them challenging – not only its digital behavior needs to be testable but it is also important to ensure that they MFs work and that downstream logic tolerates arbitrarily long metastability resolution times. There is clearly a gap to fill in the state of the art to ensure the addressing of these MUTEX issues.

This paper starts with two classical MUTEX designs: *i*) the full-custom [10], [11]; and *ii*) the standard-cell-based [12]. It demonstrates these designs do not provide coverage to all stuck-at-faults and proposes new MUTEXes, not only to cover all such faults but also to enable testing metastability. Circuits target a 65 nm bulk CMOS technology. Assessing the costs of adding testability occurs by comparing power, area and metastability resolution times. The analysis of metastability resolution times is particularly important, as a recent paper demonstrated that naively adding design-for-test features to other components such as synchronizers can dramatically decrease mean-time-between-failures (MTBF) [13]. The analyses rely on extensive electrical simulation and MetaACE [14], a commercial tool for computing MTBF. The experiments comprise process, voltage and temperature variation analyses. Results indicate that the proposed testable full-custom design does not significantly compromise MTBF and introduce penalties of 44% in area, 25% in dynamic power and 97% in leakage power, while improving delay by 9%.

The remaining of this paper comprises four sections. Section II introduces the basic behavior of the MUTEX and of the two baseline implementations. Next, Section III explores the testability issues of the baseline MUTEXes and introduces two new designs ensuring full coverage of stuck-at faults which enable testing of the MFs. Section IV presents the employed experimental setup and discusses the obtained results. Finally,

§ Peter A. Beerel is also a Chief Scientist, Technology Development at Intel, Calabasas, CA 91302.

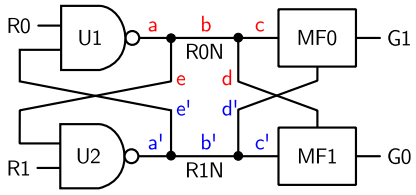


Fig. 1. Basic MUTEX composed by two cross-coupled NANDs and a metastability filer (MF), adapted from [2].

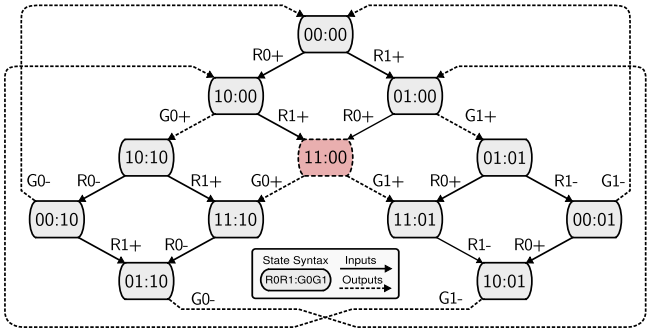


Fig. 2. State diagram of a MUTEX, based on Greenstreet [2].

Section V addresses some discussions, conclusions and future work.

II. The Traditional MUTEXes

A. Behavioral Analysis of Baseline MUTEXes

A MUTEX is a circuit that guarantees mutual exclusion among requests from entities sharing access to one resource. As Figure 1 shows, the simplest MUTEX has two request inputs (R0 and R1) and two grant outputs (G0 and G1). Its internal organization usually comprises two cross-coupled NANDs (U1 and U2) that receive the input requests, followed by a metastability filter (MF). The latter consists in blocks MF1 and MF2, their connections to the output of the cross-coupled NANDs and to the MUTEX primary outputs. Figure 2 shows the discrete behavior of the MUTEX using a state diagram. In this diagram, state representation involves primary inputs and outputs in the order R0R1:G0G1. Also, solid edges represent input transitions while dashed edges represent the MUTEX behavior due to the corresponding input changes.

Assuming a starting state where R0 and R1 are at 0, U1 and U2 write a 1 on their outputs (R0N and R1N). When the MF has 1s on its inputs R0N and R1N, it sets its outputs G0 and G1 to 0, represented by the state 00:00 in Figure 2. For a transition on a single input R0/R1, with the other input remaining at 0, one of gates U1/U2 respectively writes 0 in R0N/R1N. Whenever one of the inputs of the MF (R0N/R1N) goes to 0, the MF writes 1 in the corresponding output (G0/G1). This appears, for instance, in the sequence of transitions 00:00→10:00→10:10. Once one output is at 1, a new low-to-high transition in the other input of the MUTEX will not affect the outputs. This occurs because at these states R0N/R1N is at 0, which masks low-to-high transitions on R1/R0. This happens for example, in states 11:10 and 11:01, where a state transition will only occur

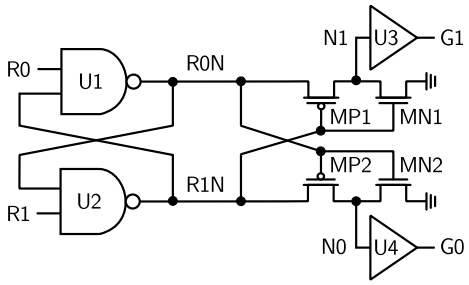


Fig. 3. Full-custom MUTEX (FC-MUTEX) design [2].

if either R0 or R1 switch to 0, in which case the MUTEX releases the asserted output, switching it back to 0.

Separate events on R0 and R1 are easily treated by the cross-coupled NAND gates. However, if both inputs switch within a time window sufficiently small, this creates a condition where both U1 and U2 are trying to switch their outputs to 0. Under such condition, the NANDs can be viewed as a loop of two inverters where both R0N and R1N will be lowered to a voltage close to $V_{DD}/2$ until one of them overpowers the other completely, switching R0N/R1N to 0, and R1N/R0N to 1. The problem is that R0N and R1N can stay at a metastable voltage for an unbounded period of time, until the NANDs resolve their outputs. If this metastability propagates to other circuits it can have unpredictable effects. This is where the MF plays a fundamental role. This component ensures that the metastable voltages do not propagate to the outputs of the MUTEX. In other words, it keeps G0 and G1 low until R0N and R1N settle to valid voltage levels (which define the logic values 0 and 1). The dashed state 11:00 marked in red in Figure 2 shows this last scenario. After R0N and R1N settle, either G0 or G1 will switch to 1. Note that the state diagram is thus non-deterministic.

B. Full Custom Design

A classic approach for implementing a MUTEX appears in Figure 3, akin to the original Seitz's proposal [10], which employs a full-custom metastability filter. Throughout this work we refer to this MUTEX as the full-custom MUTEX (FC-MUTEX). As Figure 3 shows, the MF of the FC-MUTEX has a topology similar to two cross-coupled inverters where each input and pMOS transistor drain connect to R0N/R1N and R1N/R0N, respectively. In this way, whenever R0N and R1N are at 1, the outputs will be tied to 0 by MN1 and MN2. Next, when one of R0N/R1N switches to 0, MP2/MP1 is activated, propagating the 1 in R1N/R0N to output G0/G1. In the event of metastability in R0N and R1N, meaning the pMOS transistor is supplied by a $V_{DD}/2$ voltage, the difference between R0N and R1N will be lower than the threshold voltage and MP2 and MP1 will be in the cut-off region. This avoids charging nodes N0 and N1. Furthermore, these nodes are kept low by transistors MN1 and MN2, which are partially active, in the saturation region. This scenario will hold until R0N and R1N settle, when the MF will propagate a 1 to only one of the outputs, according to the values of R0N and R1N.

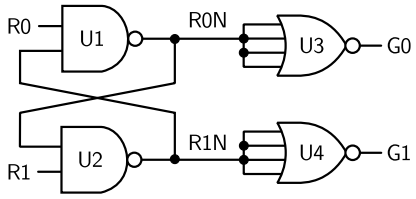


Fig. 4. Standard cell MUTEX design where the metastability filter is made by NOR gates [2].

C. Standard Cell Design

Another possible implementation is the standard cell MUTEX (SC-MUTEX). This design comprises of gates available in conventional standard cell libraries only. As Figure 4 shows, other than the two NANDs, it requires only a NOR gate with multiple inputs (more than 2) tied together. The work employs a 4-input NOR (NOR4). The NOR4 gate behaves like a skewed inverter where the voltage transfer curve shifts towards 0 for the input voltage. In other words, the switching threshold for high-to-low transitions in the output of the NOR4 reduces, while the switching threshold for low-to-high input transitions increases. This is possible due to the stacking of pMOS transistors, the parallel arrangement of nMOS transistors in the NOR4 gate and the fact that all inputs are connected together. In this case, for normal operation, when the voltage levels of R0N and R1N are above/below the voltage of a logic 1/0, the NOR4 behaves like an inverter. However, during a metastable state, the voltage on R0N and R1N will not be sufficient to switch the output to 1, due to its skewed nature, and G0 and G1 keeps at 0 until R0N and R1N settle to a valid voltage level.

III. Design for Testability

The FC-MUTEX circuit has two testability issues. First, the metastability filters in these designs cannot be tested without changing their structure. This is because forcing inputs of the MFs into metastability is the only method to test its filtering capabilities. However, no test vector can generate $V_{DD}/2$ at the inputs of the MFs for a prolonged period of time. Second, if we use the stuck-at-fault (SAF) model to evaluate fault coverage, two SAFs cannot be detected in the original MUTEX circuits. In particular, we model the FC-MUTEX at the gate-level, via Figure 1, adapted from [2]. In doing so, the MFs are defined as components with 2 inputs and 1 output. We then consider each wire segment as a potential location of a stuck-at-0 (SA0) or stuck-at-1 (SA1), yielding 28 possible FC-MUTEX SAFs. Using the checkpoint theorem [15] and fault equivalence and dominance [16], it is possible to reduce these 28 possible faults to the 18 presented in Table I.

Table I shows the SA1 fault is not detectable on nodes c and c' by any test vector, which results in a fault coverage of 90%. This is because the cross-coupled NAND gates cannot generate a $\{0, 0\}$ at their outputs that is necessary to turn on the MP1/MP2 transistors to observe the impact of these faults at the MUTEX outputs. If either of these two untestable SAFs occur, the MF may not properly filter metastability. In particular, if c is SA1 and the other input of the MF is metastable, G1 may exhibit metastability as well. In conclusion, it is necessary to detect these two SAFs to make sure the MF works.

TABLE I
Stuck-at fault analysis for FC-MUTEX.

Node	Test	Detectable	Test Vector {R0, R1}	Fault-free Output {G0, G1}	Faulty Output {G0, G1}
R0	SA0	Yes	{1, 0}	{1, 0}	{0, 0}
R0	SA1	Yes	{0, 0}	{0, 0}	{1, 0}
R1	SA0	Yes	{0, 1}	{0, 1}	{0, 0}
R1	SA1	Yes	{0, 0}	{0, 0}	{0, 1}
G0	SA0	Yes	{1, 0}	{1, 0}	{0, 0}
G0	SA1	Yes	{0, 1}	{0, 1}	{1, 1}
G1	SA0	Yes	{0, 1}	{0, 1}	{0, 0}
G1	SA1	Yes	{1, 0}	{1, 0}	{1, 1}
c	SA0	Yes	{0, 1}	{0, 1}	{0, 0}
c	SA1	No	–	–	–
c'	SA0	Yes	{1, 0}	{1, 0}	{0, 0}
c'	SA1	No	–	–	–
d	SA0	Yes	{0, 0}	{0, 0}	{1, 0}
d	SA1	Yes	{1, 0}	{1, 0}	{0, 0}
d'	SA0	Yes	{0, 0}	{0, 0}	{0, 1}
d'	SA1	Yes	{0, 1}	{0, 1}	{0, 0}
e	SA1	Yes	{1, 0}→{1, 1}	{1, 0}→{1, 0}	{1, 0}→{0, 0}
e'	SA1	Yes	{0, 1}→{1, 1}	{0, 1}→{0, 1}	{0, 1}→{0, 0}

A similar testability analysis can be performed for the SC-MUTEX design finding two similar issues. First, the filtering of metastability cannot be tested, because the inputs of the NOR gates cannot be forced into metastability for a prolonged period of time by any test vector. Secondly, SA0 faults on the inputs to the 4-input NOR gate cannot be detected.

A. Full-Custom Design for Testability

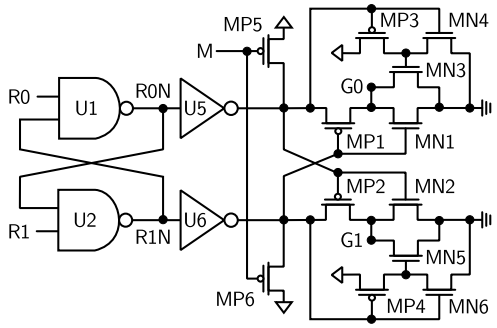
To increase fault coverage, a new MUTEX component is proposed. It is the DFT-FC-MUTEX, depicted in Figure 5(a). Two inverters, namely U5 and U6 are inserted after the cross-coupled NAND gates to generate a logic 0 at both inputs of the MFs to cover the SAFs described above, which are not covered by the FC-MUTEX. These inverters enable c (and c') to be set to 0 (when inputs R0 and R1 are 0), which sensitizes the MFs to observe if f (and f') are SA1. Note that to compensate for these added inverters, a swap of outputs G0 and G1 takes place.

A disadvantage of these two extra inverters is that when inputs of the metastability filters are 0, MP1 and MN1 of MF0 are in the cut-off state, which means G0 is floating. To solve this problem we added transistors MP3, MN3 and MN4 to the MF, to generate a strong 0 on G0, when its inputs are at logic 0. We added pMOS transistor MP5 to enable forcing the input of the MFs to metastability for a prolonged period of time, by creating a voltage divider between MP5 and the pull-down of U5. The same disadvantage occurs and is solved in the same way for the other circuit shoulder. Note that, as Figure 5(b) shows, MP5 and MP6 are modeled as pull up components PU1 and PU2, to enable the use of the stuck-at model. As Table II shows, the proposed design enables covering all SAFs.

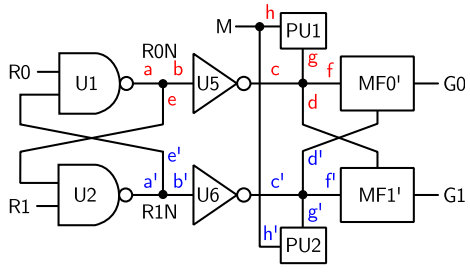
Table II also shows the logic added is also testable. In particular, all SAFs on the newly introduced signal M input and its branches can be detected. SA1 on M is tested by setting R0=0, R1=1 and M=0. Note that, to ensure coverage of this

TABLE II
STUCK-AT FAULT ANALYSIS FOR FC-DFT-MUTEX.

Node	Test	Detectable	Test Vector {R0, R1}	Fault-free Output {G0, G1}	Faulty Output {G0, G1}
R0	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
R0	SA1	Yes	{0, 0, 1}	{0, 0}	{1, 0}
R1	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
R1	SA1	Yes	{0, 0, 1}	{0, 0}	{0, 1}
G0	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
G0	SA1	Yes	{0, 1, 1}	{0, 1}	{1, 1}
G1	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
G1	SA1	Yes	{1, 0, 1}	{1, 0}	{1, 1}
b	SA0	Yes	{0, 0, 1}	{0, 0}	{1, 0}
b	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
b'	SA0	Yes	{0, 0, 1}	{0, 0}	{0, 1}
b'	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
d	SA0	Yes	{0, 1, 0}	{0, 0}	{0, 1}
d	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
d'	SA0	Yes	{1, 0, 0}	{0, 0}	{1, 0}
d'	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
e	SA1	Yes	{1, 0, 1}→{1, 1, 1}	{1, 0}→{1, 0}	{1, 0}→{0, 0}
e'	SA1	Yes	{0, 1, 1}→{1, 1, 1}	{0, 1}→{0, 1}	{0, 1}→{0, 0}
f	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
f	SA1	Yes	{0, 0, 1}	{0, 0}	{1, 0}
f'	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
f'	SA1	Yes	{0, 0, 1}	{0, 0}	{0, 1}
g	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
g	SA1	Yes	{0, 0, 1}	{0, 0}	{1, 0}
g'	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
g'	SA1	Yes	{0, 0, 1}	{0, 0}	{0, 1}
h	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
h	SA1	Yes	{0, 1, 0}	{0, 0}	{0, 1}
h'	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
h'	SA1	Yes	{1, 0, 0}	{0, 0}	{1, 0}
M	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
M	SA1	Yes	{0, 1, 0}	{0, 0}	{0, 1}



(a)



(b)

Fig. 5. Proposed DFT-FC-MUTEX (a) transistor-level and (b) block/gate-level schematics.

fault, adjustments on transistors sizing may be required for the MFs. Specifically, MP1/MP2 connect to f/f' and MN1/MN2 connect to d'/d. The nMOS transistors must be strong enough

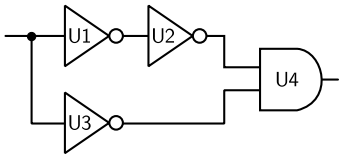


Fig. 6. Metastability detector proposed in Razor [17].

TABLE III
STUCK-AT FAULT ANALYSIS FOR SC-DFT-MUTEX.

Node	Test	Detectable	Test Vector {R0, R1}	Fault-free Output {G0, G1}	Faulty Output {G0, G1}
R0	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
R0	SA1	Yes	{0, 0, 1}	{0, 0}	{1, 0}
R1	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
R1	SA1	Yes	{0, 0, 1}	{0, 0}	{0, 1}
G0	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
G0	SA1	Yes	{0, 1, 1}	{0, 1}	{1, 1}
G1	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
G1	SA1	Yes	{1, 0, 1}	{1, 0}	{1, 1}
b	SA0	Yes	{0, 0, 1}	{0, 0}	{1, 0}
b	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
b'	SA0	Yes	{0, 0, 1}	{0, 0}	{0, 1}
b'	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
d	SA1	Yes	{1, 0, 0}	{0, 0}	{1, 0}
d'	SA1	Yes	{0, 1, 0}	{0, 0}	{0, 1}
e	SA1	Yes	{1, 0, 1}→{1, 1, 1}	{1, 0}→{1, 0}	{1, 0}→{0, 0}
e'	SA1	Yes	{0, 1, 1}→{1, 1, 1}	{0, 1}→{0, 1}	{0, 1}→{0, 0}
g	SA0	Yes	{0, 1, 1}	{0, 1}	{1, 1}
g	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
g'	SA0	Yes	{1, 0, 1}	{1, 0}	{1, 1}
g'	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
h	SA0	Yes	{0, 1, 1}	{0, 1}	{1, 1}
h	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
h'	SA0	Yes	{1, 0, 1}	{1, 0}	{1, 1}
h'	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
i	SA0	Yes	{1, 0, 1}	{1, 0}	{0, 0}
i	SA1	Yes	{1, 0, 0}	{0, 0}	{1, 0}
i'	SA0	Yes	{0, 1, 1}	{0, 1}	{0, 0}
i'	SA1	Yes	{0, 1, 0}	{0, 0}	{0, 1}
j	SA0	Yes	{1, 0, 0}	{0, 0}	{V, 0}
j	SA1	Yes	{1, 0, 1}	{1, 0}	{0, 0}
j'	SA0	Yes	{0, 1, 0}	{0, 0}	{0, V}
j'	SA1	Yes	{0, 1, 1}	{0, 1}	{0, 0}
k	SA0	Yes	{1, 0, 0}	{0, 0}	{V, 0}
k'	SA0	Yes	{0, 1, 0}	{0, 0}	{0, V}
l	SA0	Yes	{1, 0, 0}	{0, 0}	{V, 0}
l'	SA0	Yes	{0, 1, 0}	{0, 0}	{0, V}
m	SA0	Yes	{1, 0, 0}	{0, 0}	{V, 0}
m'	SA0	Yes	{0, 1, 0}	{0, 0}	{0, V}

to put the outputs to strong 0, even with a gate voltage close to $V_{DD}/2$.

With the addition of U5-6 and MP5-6, the inputs of the MFs can be set close to $V_{DD}/2$ when M = 0 and R0 = R1 = 1. We can thus test metastability filtering by observing the outputs of the MFs. In particular, we can connect each MF output to a metastability detector (MD) [17]. These detectors can be created out of standard logic and two skewed inverters, labelled U1 and U3, as Figure 6 illustrates. The closer to $V_{DD}/2$ the input is, the higher the chance it has to generate a 1 at the output of the detector [17]. If MD output goes high during the test, the MUTEX fails to filter metastability.

B. Standard Cell Design for Testability

The SC-MUTEX can also be modified to improve the coverage of all SAFs. To do so, we introduce two tristate

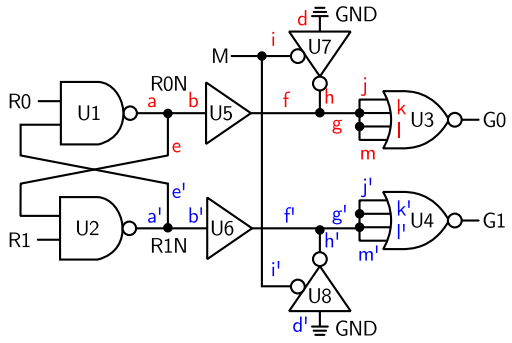


Fig. 7. Proposed standard-cell-based testable MUTEX.

inverters, typically available in standard cell libraries, as shown in Figure 7. In particular, the tristate inverters U7/U8 and the buffers U5/U6 create voltage dividers that force the inputs of the standard cell MFs to close to $V_{DD}/2$. When {R0, R1, M} = {1, 0, 0}, the output of U5 will be forced close to $V_{DD}/2$ and the NOR4 connected to G0 when operating correctly is expected to keep the outputs at logic 0. Similarly, we can test the other NOR4 gate. Single SA0 faults on inputs of NOR4 gates cannot be detected because they are a non-controlling value. However the proposed new structure with a metastability detector at the output can still potentially identify these faults. If one of the inputs is SA0 and the other input is close to $V_{DD}/2$, the output voltage of NOR4 will shift away from GND, labeled as V in Table III, which may be detected by metastability detectors at the MUTEX outputs. Thus, the new feature of forcing inputs of the NOR4s to close to $V_{DD}/2$ can help detect SA0 faults on their inputs.

IV. Experiments and Discussion

To verify the overheads introduced by the new testable MUTEXes, their design targets the STMicroelectronics 65 nm bulk CMOS technology. Explored parameters are area, power and delay trade-offs through electrical simulation of the circuits extracted post-layout. Next, there is an MTBF exploration of the new designs, comparing these to the original ones.

A. Testable MUTEXes Design

The first step of our experimental work was to design the circuits described in Figs. 3, 4, 5(a) and 7. To enable a fair comparison, we adopted the following guidelines during the design of the MUTEXes:

1) The cross-coupled NAND gates (U1 and U2) that are common in all designs use the same transistors sizes;
2) The outputs of NANDS in all designs have similar capacitances, *i.e.* nodes R0N and R1N have equalized loads;
3) All MUTEXes have the same driving strength, *i.e.* are capable of charging/discharging a given load in the same period of time.

Guidelines 1 and 2 guarantee that metastable events are caused in a similar way in all designs, due to the same NAND gates and similar loads in their outputs. To ensure guideline 1, the NAND gates are those available in the technology standard cell (core) library, for all designs. For guideline 2, the base

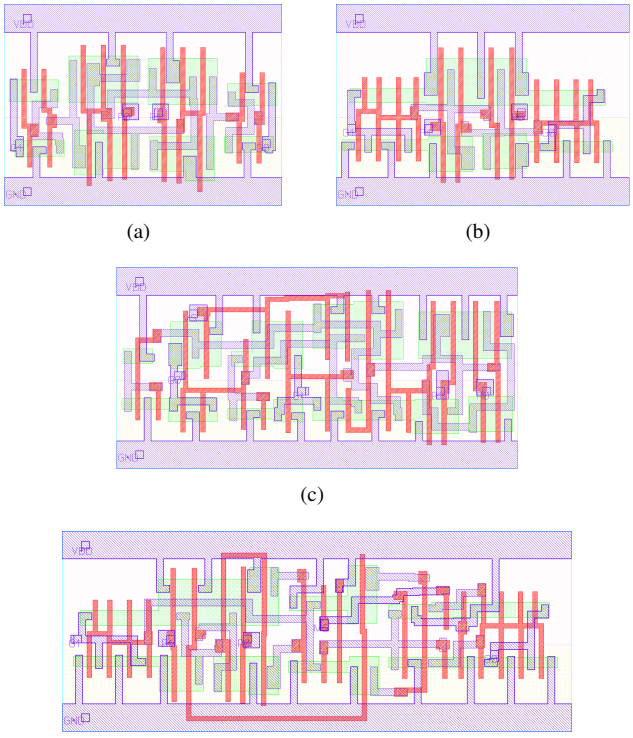


Fig. 8. Layout of (a) FC-MUTEX, (b) SC-MUTEX, (c) DFT-FC-MUTEX and (d) DFT-SC-MUTEX.

is the FC-MUTEX, which is more problematic in this case, because the load in R0N and R1N can vary depending on the load in the output of the MF. The choice was to use an MF available in the ASCEnD library [18] [19], adding output buffers as depicted in Figure 3. The ASCEnD library supports the same target technology. Measurements of the resultant capacitance in the outputs of the cross-coupled NAND gates (nodes R0N and R1N) served as a guide to design the remaining of the circuits. For the SC-MUTEX, we picked NOR4 gates from the core library that created similar loads in nodes R0N and R1N. However, this caused a mismatch in the driving strength of the designs, which was corrected by adjusting the output inverter of the output buffers of the FC-MUTEX to match the strength of the NOR4 gates. This ensures the respect of guideline 3. For the testable designs this process was facilitated by the fact that they employ inverters/buffers in their internal nodes just after the cross-coupled NANDs, as shown in Figs. 5 and 7. After designing the MUTEXes, we used the ASCEnD flow [19] to automatically generate their layouts and extract parasitics. Figure 8 show the most relevant layout layers for all MUTEXes.

B. Area, Power and Performance

Table IV summarizes the MUTEXes characteristics, pointing out their area, power and performance trade-offs. Area was collected from the designed layouts, while electrical simulation of the circuits allowed to obtain power and performance figures. In the simulation environment all circuits employed

TABLE IV
ENERGY, LEAKAGE POWER, DELAY AND AREA TRADE-OFFS FOR MUTEXES.

Designs	Avg EPT	Avg Leak. Power	Avg t_{pd}	Avg Tran. Delay	Area
FC-MUTEX	4.81 fJ	117.83 nW	48.77 ps	16.64 ps	9.36 μm^2
DFT-FC-MUTEX	5.99 fJ	231.59 nW	44.53 ps	32.64 ps	13.52 μm^2
Overhead	25%	97%	-9%	96%	44%
SC-MUTEX	3.02 fJ	71.45 nW	61.09 ps	70.96 ps	9.88 μm^2
DFT-SC-MUTEX	5.76 fJ	163.81 nW	86.33 ps	72.80 ps	17.16 μm^2
Overhead	91%	130%	41%	2%	74%

typical transistors operating at nominal voltage (1.0V) and temperature (25°C). Also, the inputs of the MUTEXes were always driven by a buffer, to allow a more realistic waveform on their inputs, and assumed the same output load, a fanout-of-4 for all designs. Simulation exercises all timing arcs (i.e. all transitions in the inputs that cause a transition in the output), and all static states (those where the circuit is stable for a given set of inputs). We collected the energy consumed for each timing arc and measured the average energy per transition (EPT). We also measured the leakage power for each static state as the average current in the power source multiplied by VDD and computed the average of these values throughout all states. Propagation and transition delays were also measured for each timing arc and we computed the average of these values for comparing the circuits.

As Table IV shows, when compared to the FC-MUTEX, the DFT-FC-MUTEX presents an overhead of 25% in average EPT, 97% in average leakage power and 44% in area. These overheads are caused by the additional transistors, that contribute to increase area, power and internal capacitance, leading to overheads in EPT. The measured average propagation delay was very similar for these designs, in fact it was slightly improved, by 9%, for the DFT-FC-MUTEX. The reason for this reduction is that the output buffer of the original FC-MUTEX could be removed for this design, since it required an internal inverter that enabled isolating the capacitance of the nodes susceptible to metastability. The overhead of 96% on the average transition delay, on the other hand, can be justified by the transistors added to the MF to ensure strong 0s. Accordingly, they increase parasitic capacitances in that node and contribute to increase the transition delay.

For the semi-custom designs (SC-MUTEX and DFT-SC-MUTEX), overheads in EPT, leakage power and area are more substantial – 91%, 130% and 74%, respectively. This is due to the fact that the designs relied only on cells available in the core library, which reduces the design optimization opportunities. Propagation delay also increases by 41%, due to the added buffers (check gates U5 and U6 in Figure 7). Transition delay, on the other hand, remains practically the same, increasing by only 2%. This is because the NOR4 used as an MF was the same in both designs.

C. MTBF Analysis

In synchronizers, MTBF is a common metric for assessing reliability. This metric enables understanding how often these circuits will have a synchronization failure caused by metastability events. MTBF can also be of importance when measuring system performance of a system composed by MUTEXes. Smaller MTBFs mean that more often the MUTEXes take long time to resolve, which impacts the system performance. As discussed in [1], [20], the MTBF can be computed by:

$$MTBF = \frac{e^{t_s/\tau}}{t_w f_c f_d}, \quad (1)$$

where τ is the resolution time constant, t_s is the settling time in which metastability should resolve to a valid logic value, t_w is the time window during which the synchronizer is vulnerable to metastability and f_c and f_d are respectively the clock and data rates. Beer et al. exploit available models to compute the MTBF for synchronizers in [21].

In synchronizers, the t_s can be easily defined using the information about the clock signal that controls them. However, in MUTEXes there is no strict limit for the settling time, as these components are typically employed in asynchronous designs and are not constrained by a clock signal. Yet, it is possible to use MTBF as a performance metric, by defining the t_s as N times the nominal propagation delay (t_{pd}) of the MUTEX. In this way, Equation (1) provides a metric for understanding how often these circuits will present delays longer than nominal, rather than failing. Accordingly, we can define f_c and f_d as the expected rates of the two asynchronous inputs R0 and R1. The values of τ and t_s can be extracted from the circuit through specific method, as [20] discusses.

As can be observed in Equation (1), MTBF is very sensitive to τ and t_s due to the exponential dependence on these parameters. In this way, approximating τ should be carefully done. In this work we rely on MetaACE [14], a tool from Blendics specifically designed to address the occurrence of metastability in logic circuits. Internally, MetaACE uses *HSpice* and *Matlab* to improve the accuracy of the analysis as described in [22].

Table V shows the computed MTBF values for the MUTEXes, considering t_s values of 1.5, 2, 3, 5 and 10 times t_{pd}, where t_{pd} is the nominal average propagation delay of the MUTEX under analysis. The values used for $f_c = f_{R0}$ and

TABLE V
VARIATION OF MTBF FOR SEVERAL VALUES OF t_s (TT CORNER, 1.0V AND 25°C, $f_c = 200$ MHz AND $f_d = 133$ MHz).

		MTBF (years) when $t_s = N \times t_{pd}$				
Designs	t_w	$N = 1.5$	$N = 2$	$N = 3$	$N = 5$	$N = 10$
FC-MUTEX	60.0e-12	2.2e-11	1.8e-10	1.2e-08	5.2e-05	6.7e+04
DFT-FC-MUTEX	61.2e-12	1.7e-11	1.3e-10	7.3e-09	2.4e-05	1.5e+04
SC-MUTEX	54.5e-12	1.0e-10	1.3e-09	2.4e-07	7.3e-03	1.2e+09
DFT-SC-MUTEX	40.0e-12	3.1e-09	1.2e-07	1.7e-04	3.4e+02	2.0e+18

TABLE VI
ANALYSIS OF THE CHANGES IN τ (ps) DUE TO P, V, AND T VARIATIONS.

	τ (ps) for 1.0V, 25°C varying process (P)			τ (ps) for TT, 25°C varying voltage (V)			τ (ps) for TT, 1.0V varying temperature (T)		
Designs	SS	TT	FF	0.9V	1.0V	1.1V	-55°C	25°C	120°C
FC-MUTEX	14.3	11.6	9.64	13.9	11.6	10.3	9.81	11.6	13.7
DFT-FC-MUTEX	13.9	10.9	9.12	13.5	10.9	9.78	9.48	10.9	13.0
SC-MUTEX	14.7	11.8	9.83	14.1	11.8	10.4	9.94	11.8	13.9
DFT-SC-MUTEX	14.6	11.8	9.84	14.1	11.8	10.5	10.0	11.8	13.9

$f_d = f_{R1}$ ratios are set to 200 MHz and 133 MHz respectively, as a point of comparison between testable and non-testable MUTEXes. As the table shows, compared to the FC-MUTEX, the DFT-FC-MUTEX reduced MTBF by 23%, 28%, 39%, 54% and 88% for a t_s of 1.5, 2, 3, 5 and 10× t_{pd}, respectively. Fortunately these overheads are on rare events and their impact at system level is negligible, unless the number of MUTEXes in the system is very high. As an example, while in the FC-MUTEX the average propagation delay would be 10× the nominal every 67,000 years, in the DFT-FC-MUTEX it would happen every 15,000 years. For the semi-custom designs, refer to Table V. Comparing the DFT-SC-MUTEX to SC-MUTEX, MTBF increased in all cases. This means that the proposed testable design will less often present long delays when compared to the original SC-MUTEX. Overall, these experiments indicate that for both standard-cell and full-custom cases, the proposed testable MUTEXes yield small impact on system performance.

Process, voltage and temperature (PVT) variations can have a negative effect on the characteristics of CMOS devices and can change τ. Because MTBF is very sensitive to variations in this parameter, it is important to verify the robustness of the proposed testable MUTEXes to such variations as well. Voltage and temperature variations can be analyzed using MetaACE, as the tool allows the designer to configure these parameters. Table VI shows the results for an analysis considering three scenarios: *i*) fixed nominal voltage and temperature varying the process assuming three corner cases: slow-slow (SS), typical-typical (TT) and fast-fast (FF); *ii*) fixed TT transistors models operating at 25°C and varying voltage, 0.9V, 1.0V and 1.1V; and *iii*) fixed TT transistors models operating at 1.0V and varying temperature to -55°C, 25°C and 120°C. These values enable analyzing separately the effects of temperature and voltage variations and were selected from the corner case models of the libraries available in the design kit of the target technology.

From Table VI, scenario i, it is possible to note that variations in process can reduce the τ between 32.58% and 34.38% as corners change from slow to fast for all MUTEXes. For scenario ii, observe the similar variations in τ for both the FC-MUTEX and the DFT-FC-MUTEX as voltage increases from 0.9V to 1.0V, respectively 26% and 27%. The same trend can be observed for the standard-cell designs, where both present a variation of roughly 26%. For scenario iii, the FC-MUTEX displays a variation of 40% as temperature varies from -55°C to 120°C. The DFT-FC-MUTEX shows a similar variation, 37%. For the SC designs, both presented similar variations, of roughly 40%. Experiments indicate that the testable designs do not introduce significant overheads, even in the presence of voltage and temperature variations.

The literature on synchronizers often considers that τ is the single dominant factor to determine MTBF. Our experiments with MUTEXes imply that a MUTEX MTBF determination depend on factors t_s and t_w as well, which do vary. Factor t_s changes based on how close input events that cause metastability occur, and its influence on MTBF is also exponential, as depicted in Equation (1). Factor t_w varies across different designs. MTBF requires consideration of all three terms together, assuming f_c and f_c are fixed.

V. DISCUSSION, CONCLUSIONS AND FUTURE WORK

Despite the ubiquity and importance of MUTEXes in asynchronous designs, their testability has not been sufficiently explored in the literature. This paper proposes new structures for improving MUTEX testability, solving two problems observed

in state-of-the-art MUTEX designs: undetectable faults and untestable metastability filters. Experimental results on testable and non-testable versions of two classic implementations of the MUTEX enabled to assess the trade-offs of adding testability to these circuits and draw the following conclusions:

- This approach enables improving fault coverage, allowing 100% coverage of SA faults in full-custom designs. It introduces a new feature to test the filtering capabilities of MFs, which can be applied to different architectures of MFs, using either full-custom or semi-custom design approaches. In fact, both testable and non-testable MUTEX circuits can be built using standard-cells only and do not necessarily require full-custom approaches, as Section III-B discusses.
- The proposed structures for enhancing testability do not add significant performance overheads to the original MUTEX designs, but do increase area and power. However, no substantial changes occur in the key performance metric of these components, MTBF. Nonetheless, caution in necessary when using testable MUTEXes, because depending on how many of these components are present in a design, the small overheads in MTBF can amplify and translate to more substantial penalties in performance at the system level.
- The τ parameter changes with PVT variations, but the measured τ values for the testable designs varied in the same proportion of those values measured for the non-testable designs. Because τ is a key component for measuring MTBF, the above analyses provide some insight of how improving testability of MUTEXes impacts the sensitivity of MTBF to PVT variations. However, because other parameters also interfere in this metric, it is still early to make definite conclusions and special care is advisable in high variability environments.

To complement this work, several future directions exist to explore, which includes:

- Explore in detail the electrical effects and deep submicron phenomena that may impact the characteristics of the newly proposed MUTEXes. For instance, phenomena like glitches that may happen due to the added logic or temperature inversion effects can have a negative effect on MTBF, specially in newer technology nodes, leading to further penalties in performance.
- Scrutinize the impact of PVT variations on the MTBF of MUTEXes, and understand the trade-offs of adding testability to these circuits in environments that present high variability. A requirement to allow such scrutiny is the enhancement of the methodology adopted by MetaACE, to incorporate techniques such as Monte Carlo simulation coupled with bisection methods.
- For the semi-custom design, an assumption is the availability of tristate and 4-input NOR cells in the technology standard cell library, which constrains the generality of the design technique. Thus, exploring alternative approaches for constructing semi-custom testable MUTEXes is also a future work.
- Connect the testable designs to comprehensive (chip) test structures and generate test stimuli with conventional tools. To do so it is necessary to better explore the challenges of using metastability detectors for identifying faulty MFs.

ACKNOWLEDGEMENTS

The Authors would like to thank Blendics for furnishing us with access to MetaACE, as well as for their extensive support in its use to address our quite specific issues. This research was partially funded by a gift from Qualcomm and grants/scholarships from CAPES (process no. 2129/14-0) and CNPq (processes no. 401839/2013-3 and no. 200147/2014-5), Brazilian research funding organisms.

REFERENCES

[1] D. Kinniment, *Synchronization and Arbitration in Digital Systems*. John Wiley & Sons, Ltd, 2008.
[2] C. Yan and M. Greenstreet, "Verifying an Arbiter Circuit," in *FMCAD*, 2008, pp. 1–9.
[3] P. Beerel, R. Ozdag, and M. Ferretti, *A Designer's Guide to Asynchronous VLSI*. Cambridge University Press, 2010.
[4] D. Fang and R. Manohar, "Non-Uniform Access Asynchronous Register Files," in *ASYNC*, 2004, pp. 78–85.
[5] A. Lines, "Asynchronous Interconnect for Synchronous SoC Design," *IEEE MICRO*, vol. 24, no. 1, pp. 32–41, 2004.
[6] R. Dobkin, R. Ginosar, and I. Cidon, "QNoC Asynchronous Router with Dynamic Virtual Channel Allocation," in *NOCS*, 2007, pp. 218–218.
[7] A. Ghiribaldi, D. Bertozzi, and S. Nowick, "A Transition-Signaling Bundled Data NoC Switch Architecture for Cost-Effective GALS Multicore Systems," in *DATE*, 2013, pp. 332–337.
[8] J. Pontes *et al.*, "Hermes-AA: A 65 nm Asynchronous NoC Router with Adaptive Routing," in *SOCC*, 2010, pp. 493–498.
[9] M. Amde *et al.*, "Asynchronous on-chip Networks," *IEE Proceedings on Computers and Digital Techniques*, vol. 152, no. 2, pp. 273–283, Mar 2005.
[10] C. Seitz, "Ideas about Arbiters," *Lambda*, vol. 1, no. 1, First Quarter, pp. 10–14, 1980.
[11] ——, *Introduction to VLSI Systems*. Addison-Wesley, 1980, ch. 7 - System Timing.
[12] K. van Berkel, F. Huberts, and A. Peeters, "Stretching Quasi Delay Insensitivity by Means of Extended Isochronic Forks," in *ASYNC*, 1995, pp. 99–106.
[13] S. Yang, I. Jones, and M. Greenstreet, "Synchronizer Performance in Deep Sub-Micron Technology," in *ASYNC*, 2011, pp. 33–42.
[14] "MetaACE, a Blendics product." http://blendics.com/solutions, accessed: March 10th, 2015.
[15] D. Bossen and S. Hong, "Cause-effect analysis for multiple fault detection in combinational networks," *IEEE Transactions on Computers*, vol. C-20, no. 11, pp. 1252–1257, 1971.
[16] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Design*. John Wiley & Sons, 1990.
[17] D. Ernst *et al.*, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," in *MICRO*, 2003, pp. 7–18.
[18] M. Moreira *et al.*, "A 65 nm Standard Cell Set and Flow Dedicated to Automated Asynchronous Circuits Design," in *SOCC*. USA: IEEE, 2011, pp. 99–103.
[19] ——, "ASCEnD2: Automated Asynchronous Cells Design," in *SBCCI*, 2014, pp. 183–188.
[20] I. Jones, S. Yang, and M. Greenstreet, "Synchronizer Behavior and Analysis," in *ASYNC*, 2009, pp. 117–126.
[21] S. Beer, J. Cox, T. Chaney, and D. Zar, "MTBF Bounds for Multistage Synchronizers," in *ASYNC*, 2013, pp. 158–165.
[22] S. Yang and M. Greenstreet, "Computing Synchronizer Failure Probabilities," in *DATE*, 2007, pp. 1361–1366.