I, Hyoung-Kook Kim, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Computer Science & Engineering.

It is entitled:
Defect-oriented fault analysis of a two-D-flip-flop synchronizer and test method for its application

Student's name: Hyoung-Kook Kim

This work and its defense approved by:

Committee chair: Wen Ben Jone, PhD
Committee member: Chien-In Henry Chen, PhD
Committee member: Harold Carter, PhD
Committee member: Carla Purdy, PhD
Committee member: Philip Wilsey, PhD
Defect-oriented fault analysis of a two-D-flip-flop synchronizer and test method for its application

A dissertation submitted to the Graduate School of the University of Cincinnati in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the School of Electronic and Computing System of the College of Engineering

by

Hyoung-Kook Kim
M.S. SungKyunKwan University
June 2012

Committee Chair: Dr. Wen-Ben JONE
Abstract

This thesis presents defect-oriented fault modeling and analysis of a two-D-flip-flop synchronizer and provides a test method for its application circuits. Bridging (open) defects are injected into any possible pair of internal nodes of the synchronizer. Then, HSPICE is used to perform the circuit analysis of each defect. The major purpose of this analysis is to acquire all possible faults that might occur in the synchronizer by each injected bridging (open) defect. Simulation results show that bridging and open defects can cause the synchronizer to generate stuck-at fault, functional timing fault, pulse output fault, one-time pulse fault, internal oscillation fault, and undefined output fault. Moreover, fault behaviors of the synchronizer depend on the location and resistance value of each defect, the input signal pattern (rising and falling), the input signal application time, and the applied clock frequency. The issues of fault behavior under the consideration of process variation, and the relationship between defects and the synchronizer failure mechanisms are also discussed. After dealing with failure analysis, an asynchronous First-In-First-Out (FIFO) interface (for multi-clock domain circuits) as an application of the two-D-flip-flop synchronizer is implemented. The number of synchronizers in the asynchronous FIFO interface depends on the width of the address lines. A general test method for the asynchronous FIFO interface is proposed. The proposed general test method evolves to the several test methods to detect the observed faults of all synchronizers in the asynchronous FIFO interface. Programmable delay generation and calibration are used to accomplish the pseudo at-speed delay testing for the FIFO circuit. Results demonstrate that the fault modeling and test methods developed in this research are effective, and can greatly enhance the reliability of a circuit which contains multiple clock domains.
Acknowledgment

First I would like to express my sincere gratitude to my advisor, Dr. Wen-Ben Jone, for all his guidance, support, and constructive inputs throughout this work. He was always ready to provide his valuable advice in solving various problems and guiding the direction of my research. This work would have never been possible without his encouragement and expertise. I am greatly impressed by his profound knowledge, creative thoughts, and careful research style. What I learned from Dr. Jone is not only valuable knowledge and skills, but also the principles and styles for research. These will surely be very beneficial for me in my future career. As an advisor, Dr. Jone also develops good friendship with his students. He always offers considerate help in their lives. I would be very thankful to him for all his help to me and my family during my entire graduate study. He taught me the positive attitude toward academic research as well as life. I am very grateful for all I learned from him, both in knowledge and in life attitude. I would like to express my deep appreciation to my thesis committee members for their valuable instructions and help in guiding my thesis research. They are listed alphabetically as follows: Dr. Hal Carter, Dr. Henry Chen, Dr. Wen-Ben Jone, Dr. Carla Purdy, and Dr. Philip Wilsey. I am sincerely grateful to them for spending their valuable time reviewing this work, raising important issues, and guiding the directions of my research. I would also like to thank my friends and colleagues in the VLSI Testing and Low Power VLSI Design group for their valuable discussions and help. They are G. Jaggannagari, J. Guo, Q. Han, S. Jhaveri, R. Keerthi, J. Liu, etc. Finally, I would express my great thanks to my wife, Hyun-Jung Lee, and my little son and daughter, Taeho (Justin) Kim and Semi (Kayley) Kim, for their wonderful love in my life and their support to my study and research. I am also very thankful to my parents, brother, and sister for their everlasting love and support ever since I was a child. The love from my family is the strong drive for me to continuously strive in my future career.
# Contents

## 1 Introduction
1.1 Multi-clock domain and synchronization issue .......................... 1
1.2 Impact of technology scaling on reliability issue ........................ 3
1.3 Our work ................................................................................. 4

## 2 Background
2.1 Multi-clock domain application .............................................. 7
2.2 Metastability ........................................................................... 9
2.3 Defect-oriented analysis of sequential circuits ............................ 12
  2.3.1 Bridging defect in sequential circuits .................................... 12
  2.3.2 Open defect analysis ......................................................... 15
  2.3.3 Test method for sequential circuit with bridging and open defect ..... 17

## 3 Experimental setup
3.1 Design of two-D-flip-flop synchronizer ........................................ 19
3.2 Analysis of fault behavior of synchronizer with bridge defect ............. 21
3.3 Analysis of fault behavior of synchronizer with open defect .............. 22

## 4 Analysis of Bridging defects of a two-D-flip-flop synchronizer
4.1 Basic bridging characteristics .................................................. 23
  4.1.1 Inverter I/O direct connection ............................................. 23
  4.1.2 Inverter I/O indirect connection by a transmission gate .............. 24
  4.1.3 Inverter output signal conflict along transmission gates ............... 24
  4.1.4 Signal conflict with two directly connected inverter outputs .......... 24
  4.1.5 Summary of basic bridging characteristics .............................. 27
4.2 Analysis of fault behaviors ..................................................... 28
  4.2.1 Stuck-at fault (SAF) .......................................................... 28
  4.2.2 Functional timing fault ....................................................... 30
  4.2.3 Pulse output fault (POF) ...................................................... 35
  4.2.4 Internal oscillation fault (IOF) ............................................. 37
  4.2.5 Undefined output fault (UOF) ............................................. 41
  4.2.6 One-time pulse fault (OTP) ................................................. 43
  4.2.7 Late transition fault (TFT3) ............................................... 44
4.3 Summary of observed fault behaviors ....................................... 51
  4.3.1 Fault behavior observation for 180nm technology model ............... 51
  4.3.2 180nm technology model vs. 45nm and 65nm technology models ..... 55
4.4 Process-Voltage (PV) variations and fault models ......................... 58
5 Analysis of Open defects of a two-D-flip-flop synchronizer

5.1 Open fault characteristics on locality

5.1.1 Open defect in a transistor of a transmission gate
5.1.2 Open defect at the drain and source of an inverter
5.1.3 Open defect at the input of inverter

5.2 Analysis of fault behaviors

5.2.1 Stuck-at fault (SAF)
5.2.2 Delay output fault (DOF)
5.2.3 Delay output fault by resistive open defect (DOFR)
5.2.4 Pulse output fault (POF)
5.2.5 Undefined output fault (UOF)
5.2.6 One-time pulse fault (OTP)

5.3 Summary of observed fault behaviors

6 Summary of fault behaviors and impact on system

7 Asynchronous FIFO and test method development

7.1 The Architecture of asynchronous FIFO design

7.2 Test Scheme and Circuits

7.2.1 Test Structure
7.2.2 Test clock generation for proposed test scheme
7.2.3 Programmable delay (PD) element
7.2.4 Channel delay compensator (CH_CMP)
7.2.5 Delayed scan enable signal generator (DSE_GEN)
7.2.6 Capture clock generator (CCLK_GEN)

7.3 A calibration method for programmable delay element

7.3.1 Calibration process
7.3.2 Calibration of channel delay compensator with \( d_{\text{adjustable}} \)
7.3.3 Calibration for capture clock generator

8 Test pattern generation for proposed test method

8.1 Initialization test
8.2 Pulse output fault (POF) detection
8.3 Delay output fault (DOF) detection
8.4 Earlier-than-expected (TFT1) fault detection
8.5 Undefined output fault (UOF) detection
8.6 Summary of test methods and test procedure

9 Conclusion and future research

9.1 Contribution of this work

9.1.1 Fault modeling of two-D-flip-flop synchronizer
9.1.2 Test for two-D-flip-flop synchronizers
9.1.3 Pseudo-asynchronous at-speed testing technique

9.2 Future work
## List of Figures

1.1 Itanium die photo ................................................ 2  
1.2 Example of bridging and open defect ........................ 4

2.1 Multi-core microprocessor trend .............................. 8  
2.2 Intel Xeon® die photo ........................................... 9  
2.3 Asynchronous FIFO interface with synchronizers .......... 10  
2.4 Timing diagram for metastable state ......................... 10  
2.5 Examples for metastability calculation ..................... 11  
2.6 Examples of bridging defects causing state modification  13  
2.7 Bridging defects in transmission gate latch ................ 14  
2.8 Examples for open drain and open source .................. 15  
2.9 NMOS transistor with the floating gate ..................... 16  
2.10 Inverter with floating gate .................................... 17

3.1 Schematic of synchronizer ...................................... 19  
3.2 Voltage transfer curve of the inverter ...................... 20  
3.3 Open fault denomination ....................................... 22

4.1 Inverter I/O direct connection ................................. 25  
4.2 Inverter I/O indirect connection by a transmission gate . 26  
4.3 Signal conflict for transmission gates ....................... 27  
4.4 A direct connection of the outputs of two inverters ....... 27  
4.5 Signal conflict for a transmission gate in synchronizer with $bc26_{DFF}$ .......................... 29  
4.6 Signal conflict for two transmission gates in synchronizer with $bb22$ ........................................ 30  
4.7 TFT1 and TFT2 .................................................... 31  
4.8 Internal signal transitions of synchronizer with $bc36_{DFF}$ for logic-0 to 1 input change for different clock cycles ................................................................. 33  
4.9 Waveforms of synchronizer with $bc36_{DFF}$ for logic-0 to 1 input change at $t_{dist} = -0.9\text{ns}$ (dotted line) and $t_{dist} = -0.7\text{ns}$ (solid line) ................................. 34  
4.10 Internal state transitions of synchronizer with $bc17_{DFF}$ when input = logic-0 for different clock cycles ................................. 36  
4.11 Internal signal transitions of synchronizer with $bc17_{DFF}$ when input = logic-1 for different clock cycles ................................. 38  
4.12 Internal signal transitions of synchronizer with $bb23$ when input = logic-0 for different clock cycles ................................. 39  
4.13 Two different outputs of synchronizer with $bc27_{DFF}$ when input = logic-0 and $freq_{CLK} = 500\text{MHz}$ .................................................. 40
4.14 Waveforms of synchronizer with $bc_{27\text{DF1}}$ when $f_{CLK} = 500\text{MHz}$ for input signal and different nodes (dotted lines) ................................................................. 41
4.15 Waveforms of synchronizer with $bc_{27\text{DF1}}$ when $f_{CLK} = 250\text{MHz}$ for input signal and different nodes (dotted lines) ................................................................. 42
4.16 Synchronizer with $bc_{56\text{DF2}}$ ................................................................. 43
4.17 Internal state changes of synchronizer with $bb_{13}$ at $R = 1.07\text{k}\Omega$ for logic-0 to 1 input change and different clock cycles ................................................................. 45
4.18 Waveform of the synchronizer with $bb_{13}$ when $R = 1.07\text{k}\Omega$ for input and different nodes (dotted lines) ................................................................. 46
5.1 Open defects in the same transistor ................................................................. 62
5.2 NMOS with an open defect at the gate ............................................................. 64
5.3 Waveform of the gate with an open defect of NMOS ........................................... 64
5.4 Inverter with open input ................................................................. 66
5.5 $OI_{18}$ open defect ................................................................. 67
5.6 Synchronizer with $OI_{34}$ ................................................................. 67
5.7 Synchronizer with $OI_{23}$ ................................................................. 68
5.8 Simulation result of the synchronizer with $OI_{23}$ ........................................... 69
5.9 $OI_{14}$ open defect ................................................................. 70
5.10 Synchronizer with $OI_{64}$ ................................................................. 71
5.11 Synchronizer with $OI_{74}$ ................................................................. 72
5.12 Synchronizer with $OI_{84}$ ................................................................. 73
6.1 Difference of one-time pulse fault between bridge and open defect ............................ 77
6.2 Example circuits driven by a synchronizer ........................................................... 78
7.1 FIFO design ................................................................. 80
7.2 Address generator ................................................................. 81
7.3 $ff\ (e\ f\ generation)$ ................................................................. 82
7.4 Test structure ................................................................. 83
7.5 Test scheme ................................................................. 83
7.6 Test clock generator ................................................................. 84
7.7 Delay element ................................................................. 85
7.8 Programmable delay (PD) element ................................................................. 85
7.9 Channel delay compensator with $d_{\text{adjustable}}$ ........................................... 86
7.10 Delayed SE generator ................................................................. 87
7.11 Launch clock generator ($LCLK\_GEN$) ................................................................. 88
7.12 Capture clock generator ($CCLK\_GEN$) ................................................................. 89
7.13 Calibration flow chart ................................................................. 90
7.14 Calibration of $CH\_CMP$ with $d_{\text{adjustable}}$ ........................................... 92
7.15 Reference clock generator ................................................................. 93
7.16 Calibration of the PD element of $d_H$ ................................................................. 94
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.17</td>
<td>Calibration of the PD element of $d_T$</td>
<td>95</td>
</tr>
<tr>
<td>7.18</td>
<td>Integrated calibration circuit for CCLK_GEN</td>
<td>96</td>
</tr>
<tr>
<td>8.1</td>
<td>Initialization test</td>
<td>98</td>
</tr>
<tr>
<td>8.2</td>
<td>Pulse output fault (POF)</td>
<td>101</td>
</tr>
<tr>
<td>8.3</td>
<td>Test scheme for POF detection</td>
<td>101</td>
</tr>
<tr>
<td>8.4</td>
<td>Pulse Fault detector</td>
<td>102</td>
</tr>
<tr>
<td>8.5</td>
<td>Delay Fault detection</td>
<td>103</td>
</tr>
<tr>
<td>8.6</td>
<td>Time windows for earlier-than-expected timing fault detection</td>
<td>104</td>
</tr>
<tr>
<td>8.7</td>
<td>Earlier-than-expected timing fault detection</td>
<td>106</td>
</tr>
<tr>
<td>8.8</td>
<td>Undefined logic value fault effect</td>
<td>108</td>
</tr>
<tr>
<td>8.9</td>
<td>Voltage window detector fault circuit</td>
<td>108</td>
</tr>
</tbody>
</table>
# List of Tables

1.1 Parametric failure .................................................. 4

2.1 Example for metastability calculation ............................. 11
2.2 Possible faults of the transmission gate latch .................. 14

4.1 Category of all bridging faults ................................... 51
4.2 Fault behaviors with bridging defects in DFF1 .................. 53
4.3 Fault behaviors with bridging defects in DFF2 .................. 54
4.4 Fault behaviors with bridging defects between DFF1 & DFF2 .... 56
4.5 Summary of experimental results for a 180nm technology model .... 57
4.6 Experimental results for 45nm and 65nm technology models .... 57
4.7 PV variation parameters ........................................... 58
4.8 Voltage changes at the shorted net in Fig. 4.2 for PV variations ... 59
4.9 Fault behavior changes by PV variations .......................... 59

5.1 Voltages at the gate with open defect of PMOS transistor ....... 63
5.2 Voltages at the gate with open defect of NMOS transistor ....... 65
5.3 Voltages at the floating input of a inverter ....................... 65
5.4 Category of all open faults ......................................... 74
5.5 Fault occurrence frequencies ...................................... 74

6.1 Category of all bridging and open faults ......................... 77

8.1 Test of synchronizers in the read (write) domain ................. 100
Chapter 1

Introduction

1.1 Multi-clock domain and synchronization issue

In 1965 Gordon Moore predicted that the number of transistors on a chip will double each year, renowned for Moore’s law. Moore’s law has evolved to Dave House’s revision: computer performance will double every 18 months [1]. To the first several years of 2000’s, major CPU manufacturers kept increasing CPU performance by increasing CPU frequency. However, this paradigm began to change after Intel Pentium 4® because the CPU frequency has reached to a plateau. Multi-core processors have become an alternative to enhance the computer performance, saying “two heads are better than one”. Presently Intel has released a 8-core Xeon® processor [2], and a 16-core Opteron® from AMD is available in market [3]. Fig. 1.1 shows the die photo of a quad-core Itanium processor implemented by a 65 nm technology with 2-billion transistors, and 12 independent frequency domains (2 core pairs, 1 system interface, 1 QR controller, 2 FBD2 domains, and 6 Intel QuickPath Interconnect domains) [4].

Besides, modern deep sub-micron (DSM) manufacturing technology brings about the advent of system on a chip (SoC) which consists of a group of heterogeneous intellectual property (IP) cores and I/O interfaces. The IP cores may use different technologies (logic, memory, analog, RF, etc). For functional and performance reasons, such a SoC application makes use of many complex clock structures and multiple clock domains [5]. It is common that IP cores in a SoC work asynchronously. As a consequence of the asynchrony in clock domains, data communication between IP cores is exposed to the problem of metastability which results from the uncertainty in timing due to the clock skew and the synchronization method between clock and data. Inevitably the issue of safe and reliable data communication has come to the fore and an elaborate synchronization method is required [6]. The 2009 ITRS roadmap [7] concludes that providing safe and reliable communication between asynchronous IP cores is one of the most challenging issues. Also, it predicts that the number of handshaking components determines the system complexity, and the percentage
of a design driven by handshaking clocking will be as high as 20% by 2012.

A promising option for handling asynchronous data communication is the deployment of globally asynchronous, locally synchronous (GALS) interfaces. Two of the most widely used GALS interfaces in industry are handshake protocol and asynchronous first-in-first-out (FIFO) interface. To implement those GALS interfaces, synchronizers are the fundamental circuit elements to remove metastability that might occur while the clock and data belonging to different clock domains are being synchronized. One of the most widely used synchronizers is the brute-force synchronizer which consists of two D flip-flops (called a two-D-flip-flop synchronizer). Even though it has the drawbacks of longer delay and nonzero mean time to failure (MTTF), it is widely used due to its design simplicity. Actually a well-designed two D-flip-flop synchronizer does not rely on any relationship between clock domains.

Including the asynchronous interface issue, the advent of SoCs and multi-core chips introduces new challenges to testability: scan test, memory test, reuse IP test, functional test, tester related, defect oriented test, etc [8]. As a promising solution, IEEE Std 1500 which is a scalable standard architecture for enabling test reuse and integration for embedded cores and their associated circuits [9] was developed. For modular and standard testing, IEEE Std 1500 defines a standard model for test information that comes with every IP core, and test wrapper design which encapsulates each embedded module under test and provides test control and test data access [10].

In addition, as the operating frequency of circuits becomes faster, Logic Built-In Self-Test (BIST) be-
comes crucial. BIST can generate test stimuli, analyze test responses, and perform at-speed testing for high-speed circuits with little or no help from an automatic test equipment (ATE). However, logic BIST faces many practical difficulties, especially at-speed testing for multi-clock, multi-frequency circuits [11]. The quality of at-speed testing is being challenged by the problem that an inter-clock logic block (existing between two synchronous clocks or asynchronous clocks) is not efficiently tested or completely ignored due to the complex test control [12]. In [11–14], at-speed logic BIST test schemes for inter-clock logic between two synchronous clock domains are proposed. Also, Wang et. al. [11] introduced an at-speed logic BIST test scheme for inter-clock logic between two asynchronous clock domains.

1.2 Impact of technology scaling on reliability issue

In current deep sub-micron (DSM) technologies, the introduction of new materials and innovative manufacturing techniques raises new and previously unknown failure mechanisms, and requires special fault analysis and modeling techniques [5, 15]. Also, manufacturing process variation becomes more difficult to control than the past due to continuous scaling down of CMOS manufacturing processes. The values of process parameters such as $L_{eff}$, $V_t$, and metal line width and spacing are distributed statistically, instead of deterministically [16].

The difficulty of process variation control makes the occurrence of parametric failures high. Therefore it is almost impossible to expect defect-free IC chips in current deep DSM manufacturing process era. Parametric failures fall into two classes: (1) intrinsic (free of defects), and (2) extrinsic (presence of defects) [5]. Table 1.1 presents possible causes of each class. An intrinsic parametric failure may come from supply voltage variation, temperature variation, or process variation. The intrinsic parametric failure can cause erroneous operations for the circuit in certain circuit conditions or environmental conditions. On the other hand, an extrinsic parametric failure may occur due to resistive vias and contacts, metal mouse-bites, metal slivers, gate oxide shorts, etc. The extrinsic parametric failure can cause resistive bridging and open defects. These resistive bridging and open defects have shown to cause the largest fraction of permanent failure types in DSM technologies [17, 18]. Fig 1.2(a) and (b) show typical bridging and open defects on interconnect lines, respectively.

These resistive defects and manufacturing process variations can cause the resistivity of an interconnect to be increased or can cause a transistor to slow down. To achieve high fault coverage, at-speed delay testing by which test vectors are applied and test responses are observed at the rated-clock speed is essential to detect the resistive defects and timing failures [19]. However, there is an increasing gap between application device
Table 1.1: Parametric failure

<table>
<thead>
<tr>
<th>Type</th>
<th>Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrinsic</td>
<td>power supply level variation, temperature variation, process variation</td>
</tr>
<tr>
<td>Extrinsic</td>
<td>resistive vias and contacts, metal mouse bites, metal slivers, gate oxide shorts</td>
</tr>
</tbody>
</table>

(a) Bridging defect  
(b) Open defect

Figure 1.2: Example of bridging and open defect

speed versus manufacturing test instrumentation speed [20]. That is, the manufacturing test instrumentation itself is inadequate for the at-speed delay test. To overcome this drawback, design-for-test (DFT) and built-in-self-test (BIST) techniques must be taken into consideration as a crucial part for VLSI circuit testing.

1.3 Our work

Two-D-flip-flop (two-D-FF) synchronizers have been commonly used in industry for reliable and safe data communication between asynchronous IP cores due to their design simplicity. Even though a two-D-FF synchronizer is so simple in circuit structure, it is not free from resistive bridging and open defects as well as manufacturing process variations in the DSM CMOS manufacturing process. However, most of research efforts have concentrated on resistive defects and process variations in a latch or a D flip-flop. With the state-of-the-art CMOS manufacturing technology, a two-D-FF synchronizer is generally designed into a very small area. As a result, it is possible for the two-D-FF synchronizer to suffer resistive or complete bridging defects between a pair of its internal lines. Also, complete or resistive open defects can be located at any of its internal lines.

Moreover, it is not appropriate for a two-D-FF synchronizer to be included in a scan design, because
it is not recommended to place combination circuits between both D flip-flops due to its timing-sensitivity. We expect that the synchronizer with a defect shows different fault behaviors depending on the location of the defect, its input application patterns, its input application time, and the applied clock frequency. Thus, current scan-based test methods are not applicable to the two-D-FF synchronizer. To make matters worse, it is difficult to test the synchronizer alone because it is not natural to implement extra test circuits to test a simple synchronizer. So, an efficient way is to test the synchronizer by the assistance of its peripheral circuits. After dealing with fault modeling and testing of a two-D-FF synchronizer, an asynchronous FIFO interface is a proper application because it is widely used in industry.

This research seeks to:

1. Find all possible fault behaviors of a two-D-flip-flop synchronizer caused by resistive or complete bridging and open defects.
2. Analyze the fault behaviors observed, reveal fault mechanism, and develop their fault models.
3. Analyze the fault behaviors under the effect of process variations.
4. Implement an asynchronous FIFO interface, and develop a test method to test the synchronizers.

To the best of our knowledge, this is the first work which has performed such massive fault injection and fault simulation for the two-D-FF synchronizer. Further, until now, there is no test method available to test an asynchronous FIFO interface. We expect that this research can help understand the difficulty of testing asynchronous circuits between different clock domains, and lead to a powerful solution.

This proposal is organized as follows:

In Chapter 2, multi-clock domain applications, metastability issues, and related research to resistive bridging and open defects are briefly reviewed.

In Chapter 3, the schematic of a two-D-FF synchronizer is presented, and basic characteristics of inverters used for the synchronizer is investigated. Also, we discuss issues such as how to inject bridging and open defects into the synchronizer and how to perform circuit simulations (using HSPICE) to search their fault behaviors.

In Chapter 4, bridging defects in a two-D-FF synchronizer are thoroughly studied by presenting their basic characteristics, fault modeling, and observed fault behaviors.
In Chapter 5, open defects in a two-D-FF synchronizer are exhaustively researched by presenting their characteristics based on their locations, analyzing their fault behaviors, and summarizing their fault behaviors.

In Chapter 6, fault behaviors of both types of defects are integrated into one table, and their impact on the system is briefly discussed.

In Chapter 7, an asynchronous FIFO interface is used as an application circuit of the synchronizer design, and a general test method is proposed to test the synchronizers. Also, the required circuits for the proposed test method are implemented as well as a calibration method for the circuits.

In Chapter 8, the proposed general test method is evolved to several test methods for detecting faults of two-D-FF synchronizer observed in this research.

In Chapter 9, conclusion and future research are presented.
Chapter 2

Background

2.1 Multi-clock domain application

As Chip Multiprocessor (CMP) has become the mainstream in processor architectures, Intel and AMD have introduced their multi-core microprocessors to the PC and server markets [21]. Fig. 2.1 presents the trend of multi-core microprocessor design of Intel (Fig. 2.1(a)) and AMD (Fig. 2.1(b)). Currently, Intel has released its Xeon® with eight cores and AMD has released its Opteron® processor with 16 cores in 2011. Moreover, PlayStation 3 contains a CELL® processor with nine cores, one power processing element, and eight synergistic processing elements built by Sony-Toshiba-IBM partnership [22]. Tilera has developed a CMP with 64 cores (TILE64) [23].

As an example, we investigate the 45 nm 8-core Intel Xeon® processor briefly as shown in Fig. 2.2. The processor has three primary clock domains [24]: a core clock domain (MCLK), an uncore clock domain (UCLK) and an I/O clock domain (QCLK). Each core has an independent clock generator. Even though the cores operate at the same clock frequency, they are not synchronized with each other. There is one central clock generator for UCLK which includes the system interface and the eight LLC cache slices. For QCLK, each QPI and SMI I/O has an independent clock generator to maximize the system flexibility. So, the processor has 15 independent clock generators, which means 15 different clock domains.

While multi-core microprocessors have homogeneous cores, SoCs contain heterogeneous intellectual-property (IP) cores using different technologies (logic, memory, analog, RF, etc), and I/O interfaces operating at different frequencies. Providing fast, safe, and reliable data transfer between IP cores in the multi-core and SoC applications is one of the most challenging issues [7]. Cummings [25] has dealt with some of the hardware design, timing analysis, synthesis and simulation methodologies for multi-clock designs. Ginosar [26] has presented the cases that prevent safe synchronization, analyzed the causes of the errors, and provided the correct synchronization circuits to get rid of the design issues which can induce
Figure 2.1: Multi-core microprocessor trend
synchronization errors.

One of the widely used data communication interface for asynchronous IP cores is the asynchronous FIFO interface as shown in Fig. 2.3. Each domain has a set of synchronizers shown in the dotted box. The number of the synchronizers is determined by the width of the address lines pointing to the FIFO memory. When a write operation is performed, a new address (wptr) to be written next time is transferred to the read clock domain to determine the empty condition of the FIFO. Also, the current read address (rptr) is transferred to the write clock domain to decide the full condition of the FIFO. So, wptr and rptr are clock domain crossing (CDC) signals and they are latched by the synchronizers in each clock domain.

2.2 Metastability

Metastability in a digital system occurs when two asynchronous signals work together in such a way that their resulting output goes to an indeterminate state [27]. A sequential circuits, with latches or flip-flops, can fall into a metastable state when its data and clock violates the setup and hold time specifications of the sequential circuit. In a synchronous system, the data always has a fixed relationship with respect to the clock. However, in an asynchronous system, their relationship is not fixed, and the data varies asynchronously with respect to the clock [28, 29]. For example, in Fig. 2.3 wptr and rptr are asynchronous to rclk and wclk, respectively. An element in the metastable state remains unstable or its output approaches at some intermediate voltage level (e.g., the logic threshold voltage). Eventually, the element will resolve the metastable condition and reach to a stable output state [30–32]. Fig. 2.4 depicts the timing diagram for a
metastable state.

\[ MTTF = \frac{e^{T/\tau}}{T_w f_D f_{CLK}} \]  \hspace{1cm} (2.1)

where $T$ is the settling time window, $\tau$ the settling time constant of the element, $T_w$ the time window of susceptibility ($aperture \ time = t_{su} + t_h$), $f_D$ the data frequency, and $f_{CLK}$ the clock frequency. MTTF is designed to be much longer than the expected lifetime of a product.
Fig. 2.5 and Table 2.1 present an example circuit and its synchronization parameters and the calculated MTTFs. When the logic cloud has 4.5ns of propagation delay as shown in Fig. 2.5(a), the MTTF is equivalent to $9.26 \times 10^{19}$ years in the case of data frequency equal 50 MHz and clock frequency equal 100 MHz. When the clock frequency is 200 MHz with the same data frequency, the MTTF is decreased to 1.09 s. However, when a D flip-flop is added between DFF1 and the clock cloud as shown in Fig. 2.5(b), the MTTF is increased to $4.47 \times 10^{44}$ years and $1.67 \times 10^{17}$ years for 100 MHz and 200 MHz of clock frequencies, respectively.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{CLK}$</td>
<td>Clock frequency</td>
<td>100</td>
<td>200 MHz</td>
</tr>
<tr>
<td>$f_D$</td>
<td>Data frequency</td>
<td>50</td>
<td>50 MHz</td>
</tr>
<tr>
<td>$T_w$</td>
<td>Susceptibility time window</td>
<td>0.101</td>
<td>0.101 ps</td>
</tr>
<tr>
<td>$1/\tau$</td>
<td>Settling time constant</td>
<td>79.2</td>
<td>79.2 ps</td>
</tr>
<tr>
<td>$T$</td>
<td>Settling time window</td>
<td>5.5</td>
<td>0.5 ns</td>
</tr>
<tr>
<td>MTTF of Example 1</td>
<td>Mean time to failure</td>
<td>$2.92 \times 10^{27}$</td>
<td>1.09 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$9.26 \times 10^{19}$</td>
<td>yrs</td>
</tr>
<tr>
<td>MTTF of Example 2</td>
<td>Mean time to failure</td>
<td>$1.41 \times 10^{22}$</td>
<td>$5.28 \times 10^{24}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$4.47 \times 10^{44}$</td>
<td>$1.67 \times 10^{17}$</td>
</tr>
</tbody>
</table>

To get rid of the metastability problem for safe and reliable data transfer between asynchronous clock
domains, one of the best solutions is to fully migrate to asynchronous design methodologies. However, major design communities are reluctant to accept these methodologies because asynchronous design tools are still immature, and moving away from the mature synchronous design infrastructures is risky and costly. An approach that maintains synchronous design methodologies is to use the well-known globally-asynchronous, locally-synchronous (GALS) clocking style with proper inter-core synchronization methods [34]. As stated in [35], there is no method to avoid metastability for data transfer between asynchronous IP cores without migrating to asynchronous design. Instead, the occurrence rate can be infinitesimal by implementing the data transfer interface with synchronizers: handshake protocol and asynchronous FIFO as shown in Fig. 2.3. The brute-force synchronizer which consists of more than one D flip-flop has been widely used to implement such an interface because of the simplicity of its design, even though it has disadvantages (longer delay and nonzero probability of failure as shown in Table 2.1). In Fig. 2.3, both synchronizers in the dotted boxes are used to synchronize \( wptr \) and \( rptr \) to \( rclk \) and \( wclk \), respectively.

2.3 Defect-oriented analysis of sequential circuits

2.3.1 Bridging defect in sequential circuits

Rodriguez-Montanes and Figueras [36] have studied bridging defects affecting sequential circuits and investigated the conditions under which bridging defects involving storage elements change the state memorized in the circuit, and analyzed their dependence on transistor size ratio and bridging defect resistance. Among the different defective conditions introduced in [36], two of them may have a state modification behavior which causes the state of a memory element to have opposite logic value. Fig. 2.6 depicts these two cases. Fig. 2.6(a) presents the first case where a bridging defect is located between a node in a floating control loop of a storage element and another node in a combination logic. The voltage of the bridging can show analog values in intermediate time period, but finally will be stable to 0V or \( VDD \). Fig. 2.6(b) depicts that two nodes in two floating control loops are connected through a bridging defect. Without consideration of the resistance value of the bridging defect, the voltage of the bridging will always evolve to \( VDD \) or 0V depending on the strength of the involved transistors. However, for the bridging defect with a resistance value, the fault behavior changes since \( Nc_1 \) and \( Nc_2 \) will have different voltage values depending on the resistance value.

Moreover, Al-Assadi et al. [37] analyzed several types of latches and flip-flops at transistor level and presented their higher-level fault models. In their results, we paid special attention to the fault behavior of a transmission gate latch because it is used to implement the synchronizer for this research. Fig. 2.7(a) presents the transmission gate latch. Bridging defects are injected into the circuit by considering all possible
pairs of transistor's nodes as shown in Fig. 2.7(b).

From the logical analysis, they identified several fault models: stuck-at, feed through, logic-level non-retention, delay, parametric, and complex faults depending on the location of each injected bridging defect. Table 2.2 describes each fault type and an example of each fault type is given below:

**Stuck-at:** $f_1$ defect make D’ connected to the ground. As a result, the output of the latch is stuck-at 0.

**Feed-through:** $f_2$ bridging defect makes input D bypass the transmission gate right after it, and the input change is shown at the output of the latch directly with the propagation delay. It can cause a timing problem.

**Logic-level non-retention:** $f_3$ defect causes stuck-on for PMOS transistor 3. When the data input of the latch at the sample edge of the clock is 1, the latch becomes 0 in the latch phase because the faulty PMOS transistor changes back the state of the latch.

**Delay:** $f_4$ defect causes stuck-open for NMOS transistor 2. Because of the characteristic of PMOS transistor and the malfunction of the NMOS transistor, $V_{DD}$ to 0V input transition becomes $V_{DD}$ to $|V_{tp}|$ instead of 0V. The latch shows its output with a delay.

**Parametric:** $f_5$ defect makes D’ and Q connected unexpectedly. The voltage of the bridge becomes indeterminate.

**Complex:** Because of $f_6$ defect, whether the transmission gate is turned on or off depends on a function of Q and CLK.
(a) Transmission gate latch

(b) Possible bridging defects in transistors

Figure 2.7: Bridging defects in transmission gate latch

Table 2.2: Possible faults of the transmission gate latch

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Behavior description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-at</td>
<td>The output of the latch is stuck-at 0 or 1.</td>
</tr>
<tr>
<td>Feed-through</td>
<td>The latch becomes transparent such that the output corresponds to the input or the clock with the propagation delay of the latch.</td>
</tr>
<tr>
<td>Logic-level non-retention</td>
<td>The state of the latch becomes the opposite in the latch phase to the data input of the latch at the sampling edge of the clock.</td>
</tr>
<tr>
<td>Delay</td>
<td>The latch shows fault-free logic behavior, with increased delay.</td>
</tr>
<tr>
<td>Parametric</td>
<td>The fault behavior is indeterminate and current ($I_{DDQ}$) monitoring is required.</td>
</tr>
<tr>
<td>Complex</td>
<td>The output of the latch is a function of the clock and the input signals.</td>
</tr>
</tbody>
</table>
They expanded the fault behaviors of a latch to a master-slave flip-flop to get its fault behaviors [37]. However, their fault modeling and analysis were given without consideration of signal conflicts caused by bridging defects, the frequency of the clock, and the arrival time of each input signal. Further, none of them deals with bridging defects between each pair of internal nodes in a D flip-flop or between two D flip-flops.

2.3.2 Open defect analysis

The fault behavior analysis and testing of resistive opens in CMOS latches and flip-flops have not been thoroughly investigated. In [38,39], fault analysis has been performed for different possible locations of open defects in a symmetric latch and flip-flop. The detectability of different opens by delay testing, instead of by stuck-at fault testing, has been researched, and the result does not give high defect coverage in sequential cells.

Due to the lack of a fault model for resistive open defects in a sequential cell, we review open defects in a transistor and an inverter. In [40] a deterministic approach is presented to predict the fault occurrence probability of floating gate defects by taking into account the process defect statistics and mask layout data. Also, it classified the open defects of a floating gate into three types: (1) open causes floating gates of p-channel transistors, (2) open causes floating gates of n-channel transistors, and (3) open causes floating gates of both p-channel and n-channel transistors. Open fault modeling through the electrical and functional analysis of a transistor is introduced in [41,42].

Fig. 2.8 presents examples for open drain and open source in a transistor. A transistor with open drain or open source behaves like a stuck-at fault, because its output is disconnected from $V_{DD}$ or the ground as shown in Fig. 2.8 [41].

![Figure 2.8: Examples for open drain and open source](image)
However, it is not easy to determine the fault model of an open gate, because it has different fault behaviors depending on $V_{DS}$ of the faulty transistor. Fig. 2.9(a) shows a n-channel transistor with an open gate. Because the gate is open (disconnected to a logic source), the gate-source voltage is:

$$V_{GS} = \frac{C_1}{C_2 + C_{gb}} \cdot V_{DS}$$

(2.2)

So, $V_{GS}$ depends on $V_{DS}$. Results of [41] demonstrated that the transistor with an open gate:

1. is off for $V_{DS}$ much less than $V_{TH}$ which is the threshold voltage of the transistor.
2. conducts a constant amount of small current for $V_{TH} < V_{DS} < \alpha \cdot V_{TH}$.
3. conducts an amount of current proportional to $(V_{GS} - V_{TH})^2$ for $V_{DS} > \alpha \cdot V_{TH}$.

An approximate value of $\alpha$ can be found from Equation 2.3:

$$\alpha = \frac{2/3WL}{C_{ox}} + 2$$

(2.3)

where $W$ and $L$ are the length and width of the transistor channel, and $C_{ox}$ and $C_{gdo}$ are the gate capacitance and the gate-drain overlap capacitance, respectively. Fig. 2.9(b) depicts $V_{GS}$ and $I_D$ with respect to $V_{DS}$. Even though the gate is open, when $V_{DS}$ is larger than $\alpha \cdot V_{TH}$, current $I_D$ flows significantly.

![Figure 2.9: NMOS transistor with the floating gate](image)

Renovell et. al. in [42] showed that a floating gate transistor is influenced by its topological environment, and the equivalent gate-to-source voltage of the floating gate transistor depends on the surrounding potential of metal lines and the drain-to-source voltage of the floating gate transistor itself. In [43,44], the authors
demonstrated that the oxide thickness is reduced to below a few tens of Å in the DSM technology era, and this causes the gate tunneling leakage strongly impacts the behavior of the defect circuits with full open defect. Fig. 2.10 shows an inverter with floating gate. Equation 2.4 shows how to compute the inverter’s voltage at the input with an open defect. \( C_{a1} \) and \( C_{a0} \) are the capacitances between the floating input and its neighboring nets which have logic 1 or logic 0. So, \( C_{a1} \) and \( C_{a0} \) are pattern-dependent. \( C_{VDD} \) and \( C_{GND} \) are the capacitances between the floating node and the power and ground rails, respectively. \( C_{int}(1) \) and \( C_{int}(0) \) are the internal capacitances inside the driven inverter. It is important to point out that \( C_{int}(1) \) and \( C_{int}(0) \) are not fixed, but directly depend on voltage \( V_{in} \). Since the trapped charge at the floating node is unknown, \( V_{trap} \) is unknown, too.

\[
\begin{align*}
C_0 &= C_{GND} + C_{a0} + C_{int}(0) \\
C_1 &= C_{VDD} + C_{a1} + C_{int}(1) \\
V_{in} &= \frac{C_1}{C_0 + C_1} \cdot V_{DD} + V_{trap}
\end{align*}
\]  

(2.4)

![Figure 2.10: Inverter with floating gate](image)

### 2.3.3 Test method for sequential circuit with bridging and open defect

Resistive and complete bridging and open defects usually lead to unexpected current path between the power supply and ground. When the circuit is not switching and inputs are held at static values (which is known as quiescent states), such bridging and open defect will cause the current to increase by orders of magnitude. \( I_{DDQ} \) testing is a well-known test method which measures the supply current (\( I_{DD} \)) in quiescent
state to determine the presence of such a defect. Rodriguez-Montanes and Figueras [36] provided their $I_{DDQ}$ testability results for sequential circuits with bridging defects. Metra et al. [45] analyzed the flip-flop testability with respect to resistive bridging faults. They considered functional testing and $I_{DDQ}$ testing techniques to calculate the fault coverage for a flip-flop with resistive bridging faults. Further, they found a few faults that produce excessive delay and escape from functional and $I_{DDQ}$ testing. However in a DSM manufacturing process, the background leakage current becomes much higher and less predictable by using $I_{DDQ}$. This makes it hard to differentiate a low leakage circuit with a defect from a naturally high leakage circuit.

Resistive opens in CMOS latches and flip-flops were considered, and a design for testability approach with the conditions for input sequences supported by scan chains to detect them were discussed in [38,39]. F. Yang et al. [46, 47] proposed new scan-based test methods targeting stuck-at, stuck-on, and stuck-open faults inside the scan cells by shifting in a sequence of several types of 0-1 patterns at a specific clock frequency and checking the shifted-out data. However, it is not recommended to apply scan testing for the two-D-FF synchronizer under consideration in this research, because it is timing-sensitive. By adding combination circuits for scan design to the synchronizer, the timing characteristics of the synchronizer is changed, and this causes the MTTF of the synchronizer to vary too.
Chapter 3

Experimental setup

3.1 Design of two-D-flip-flop synchronizer

To analyze all possible bridging defects in a synchronizer, we first determined the schematic of a D flip-flop to implement the synchronizer under test. We selected a simple and widely used positive edge-triggered D flip-flop which consists of four transmission gates and four inverters [48, 49]. Fig. 3.1 gives the schematic of the synchronizer which consists of two D flip-flops. DFF1 is the first D flip-flop and DFF2 is the second D flip-flop of the synchronizer. Each signal line is assigned a number, while each inverter (transmission gate) is assigned a name (e.g., i1 and t1).

\[
\begin{align*}
W_p/L_p &= 25\lambda/2\lambda \\
W_n/L_n &= 10\lambda/2\lambda \\
W/L &= 4\lambda/2\lambda
\end{align*}
\]

We used a TSMC 180nm technology model and 45nm and 65nm predictive technology models [50] for this experiment. According to the PMOS and NMOS mobilities of the TSMC 180nm technology model, we designed each inverter with \(W_p/L_p = 25\lambda/2\lambda\) (\(1\lambda = 90\text{nm}\)) and \(W_n/L_n = 10\lambda/2\lambda\) and each transmission gate with \(W/L = 4\lambda/2\lambda\), respectively. Fig. 3.2 shows the voltage transfer curve (VTC) of the implemented inverter. The \(V_{DD}\) used in the experiments is 1.8V and the logic threshold voltage \(V_{LTH}\) is 0.874V for the 1.8V \(V_{DD}\).
Figure 3.2: Voltage transfer curve of the inverter
3.2 Analysis of fault behavior of synchronizer with bridge defect

We performed experiments on the synchronizer implemented using a 180nm TSMC technology model by injecting bridging defects. We intended to consider all cases of bridging faults in the synchronizer: bridging faults in a D flip-flop (represented by $bcxy$) and bridging faults between two D flip-flops (represented by $bbxy$). In the following discussions, each fault is represented by $bcxy$ or $bbxy$ where $x$ and $y$ denote two nodes that are bridged, and $bc$ ($bb$) stands for bridging in a cell (between two cells). Further, one defect can be represented by different names, e.g., $bc17_{DF2}$ is exactly the same as $bb67$ and in such cases only the results for the “bc” notation is considered. We expected that the synchronizer with a bridging fault shows different fault behaviors depending on input pattern arrival time as well as input pattern type. So, for each bridging defect, we applied two input patterns, rising and falling signal transitions, at several different time positions within a clock cycle period which is $2\,\text{ns}$ in this experiment. To establish the probability for each fault to occur in a given synchronizer, layout information is required. To make it general, instead of using a specific layout, we performed an experiment with the synchronizer injected 0\,$\Omega$ bridging faults (to maximize the fault effect) to each pair of circuit nodes to obtain its fault behavior. According to the fault behaviors observed, we can categorize the bridging faults and analyze their fault behaviors to extract the common fault characteristics for defect understanding. With a full defect understanding, we then take the resistance of each bridging fault into account.

Given a bridging defect, to look for the resistance value from which the synchronizer works properly, we used the bisectional Pass/Fail analysis function provided by Synopsys HSPICE\textsuperscript{®} simulator. The bisection function returns a resistive value (called critical resistance [51]) from which the synchronizer works properly, though the bridging defect is existent. Note that a large value of resistance is similar to the condition in which the corresponding bridging defect is non-existent. Then, we chose three different resistive values which are computed by dividing the critical resistance by four. We simulated the synchronizer with those three resistances (besides the zero-resistance bridging) to further explore their possible new fault behaviors. Exhaustively, there are 78 pairs of bridges in the circuit and each pair is given four resistance values for simulation. Totally, we injected 312 faults for this experiment. Unlike single stuck-at faults, there are no equivalent (identical) faults for bridging defect analysis. However, they can be categorized into several fault types as summarized in Table 4.1 later. Finally, we performed the same experiment on the synchronizer with 0\,$\Omega$ bridging defects implemented by 45nm and 65nm predictive technology models.
3.3 Analysis of fault behavior of synchronizer with open defect

We intended to consider all cases of open defects in the synchronizer: open defects in a transmission gate, and open defects in an inverter. Fig. 3.3 presents the rule of denomination of open defects in the transmission gate and the inverter. For example, $OT_{pq}$ stands for an open fault at position $q$ in the transmission gate $p$. $OI_{xy}$ means an open defect at position $y$ in inverter $x$.

![Figure 3.3: Open fault denomination](image)

In the same analysis as bridging defects, we expected that open defects in the synchronizer show different fault behaviors depending on input signal arrival time as well as input signal pattern. So, for each open defect, we applied two test patterns, rising and falling signal transitions, at several different time points within a clock cycle period which is $2 \text{ns}$ in this experiment. We first performed an experiment with the synchronizer injected open defects with infinite resistance ($10G\Omega$) to maximize the fault effect. Then, according to fault behaviors observed, we categorized the open faults and analyzed their fault behaviors to extract their common fault characteristics for defect understanding.

Then, for each open defect, we used the *bisectional Pass/Fail* analysis function provided by HSPICE® simulator to look for the resistance (called *critical resistance* [51]) value (from zero resistance) to which the synchronizer works properly. Finally, for each open defect $f$, we added 15 different resistance values which are evenly distributed from $0\Omega$ to $1G\Omega$ in logarithm scale one by one to the sensitive resistance of $f$ to further explore their new fault behaviors.
Chapter 4

Analysis of Bridging defects of a two-D-flip-flop synchronizer

4.1 Basic bridging characteristics

Before we analyze the fault behaviors of the synchronizer, there is a need to investigate several basic bridging structures and characteristics that will be used in various fault injection experiments.

4.1.1 Inverter I/O direct connection

Fig. 4.1(a) shows an example in which a bridging defect connects the input/output of an inverter. When the clock is high ($\text{CLK} = 1$ and $\text{CLKB} = 0$) and the input is logic-0, as shown in Fig. 4.1(b), the latch is in the memory mode and the driving inverter a1 is disconnected from the latch. Because the $V_{GS}$ and $V_{DS}$ for all PMOS (NMOS) transistors of both inverters (a2 and a3) in the latch are the same, the transistors are in the saturation region due to Eq. 4.1 (conditions for a transistor to work in saturation mode), and the voltage of the shorted net is $V_{LTH}$ (0.874V) which is the inverter logic threshold voltage. Fig. 4.1(c) gives the case where the clock is low ($\text{CLK} = 0$ and $\text{CLKB} = 1$) and the input is logic-0, and Fig. 4.1(d) shows its corresponding circuit diagram. When the clock is low, inverter a1 instantaneously supplies charge to inverter a2 through transmission gate TG1. It causes the NMOS of a2 to be turned on, which is in the saturation region and the PMOS of a2 turned off (Fig. 4.1(d)). However, both ports of TG1 are driven by two different voltages, 1.707V and 0.987V. The NMOS of TG1 is working in the saturation region, the PMOS in the linear region (Eq. 4.1), and TG1 shows a resistive effect. The resistances of the PMOS of a1, TG1, and the NMOS of a2 are involved in determining the voltages of the signal nets. If the input is logic-1 and the clock is low in Fig. 4.1, the output voltage of a1 is 0.140V and the voltage of the shorted net is 0.644V.
\[ V_{GS} - V_{DS} \leq V_{TN} \text{ (for NMOS)} \] (4.1a)
\[ V_{GS} - V_{DS} \geq V_{TP} \text{ (for PMOS)} \] (4.1b)

4.1.2 Inverter I/O indirect connection by a transmission gate

Fig. 4.2(a) shows that the outputs of a1 and a2 have a direct connection, and a2 shows an inverter I/O bridging through transmission gate TG1. When the input is logic-0 (1) and the clock is high, as shown in Fig. 4.2(b), a1 sets (resets) the state of the latch. Fig. 4.2(c) shows the case where the clock is low and the input is logic-0, and Fig. 4.2(d) is its corresponding circuit diagram. When the clock is low, a1 begins to charge the input of a2. As a result, the NMOS of a2 is turned on, however it is in the saturation region because its \( V_{GS} \) and \( V_{DS} \) are equal (Eq. 4.1). The voltage of the shorted net is determined by the resistances of the PMOS of a1 and the NMOS of a2, which is 1.387V in this case. If the input is logic-1, the voltage of the shorted net is 0.352V.

4.1.3 Inverter output signal conflict along transmission gates

A bridging defect can cause signal conflict for transmission gates which are turned on in the synchronizer. There are two cases for consideration as shown in Fig. 4.3. Fig. 4.3(a) shows the case where both ports of a transmission gate, TG1, are driven by two different logic values from two inverters. In this case, even though both ports of TG1 show voltage drop and rise, both transistors of the transmission gate are working in the saturation region and TG1 shows a large resistance effect.

Fig. 4.3(b) shows the case where there is a conducting path passing through two transmission gates, TG2 and TG3, when both are on. In this case, the voltages of both ports of TG2 are 1.609V and 0.561V respectively, and both transistors of TG2 are in the saturation region. The voltages of both ports of TG3 are 0.561V and 0.132V, respectively. The NMOS of TG3 is in the linear region, while its PMOS in the saturation region. So, if the signal net between TG2 and TG3 is fed to an inverter, the output of the inverter becomes logic-1.

4.1.4 Signal conflict with two directly connected inverter outputs

A bridging defect can cause a direct connection between the outputs of two inverters as shown in Fig. 4.4. When two inverters have different logic values, the voltage of the shorted net is determined by the
(a) An example

\[ V_{\text{LTH}} = 0.874V \]

(b) \( \text{CLK} = 1 \) & \( \text{Input} = \text{logic-0} \)

(c) \( \text{CLKB} = 1 \) & \( \text{Input} = \text{logic-0} \)

(d) Circuit diagram

Figure 4.1: Inverter I/O direct connection
(a) An example

(b) $\text{CLK} = 1$ & $\text{Input} = \text{logic-0}$

(c) $\text{CLKB} = 1$ & $\text{Input} = \text{logic-0}$

(d) Circuit diagram

Figure 4.2: Inverter I/O indirect connection by a transmission gate
resistances of the transistors which are turned on. For the inverters used to implement the synchronizer

circuit, the on-resistance of NMOS is larger than that of PMOS. The voltage of the shorted net is 0.997V.

Figure 4.4: A direct connection of the outputs of two inverters

4.1.5 Summary of basic bridging characteristics

In this section, we investigated several basic bridging structures and characteristics that are useful to help

analyze and understand fault behaviors of the synchronizer under test. We can summarize the basic bridging

characteristics into five cases and the types of fault behaviors that can result from them. Note that we

performed this analysis using a TSMC 180nm technology model with $V_{DD}$ equal 1.8V by injecting 0 Ω

bridging defects.

1. The input of an inverter is directly connected to its output (Fig. 4.1). The voltage of the shorted net

depends on whether the transmission gate connected to the input of the inverter is turned on or not.

If the transmission gate is turned off, the voltage becomes $V_{LTH}$ regardless of its input (Fig. 4.1(b)).

Otherwise, if logic-1 (logic-0) is propagated through the transmission gate, the voltage of the shorted

net becomes 0.987V (0.644V) which is Weak-1 (Weak-0) as shown in Fig. 4.1(c). This bridging

structure can cause stuck-at and undefined output faults for the synchronizer under test.
2. When the output of an inverter drives another inverter which has an I/O indirect connection by a transmission gate (Fig. 4.2(a)), the voltage of the shorted net depends on the driving inverter. If the input of the driving inverter is logic-0 (logic-1), the logic value of the shorted net is logic-1 (logic-0) or Weak-1 (Weak-0) as shown in Fig. 4.2(b) and 4.2(c), respectively, depending on whether the transmission gate TG1 in Fig. 4.2(a) is turned off or on. This bridge can result in timing faults (early transition, inverted output, and inverted late transition faults) for the synchronizer.

3. If both ports of a transmission gate have opposite logic values and it is turned on (Fig. 4.3(a)), then both ports keep their logic values even though there are slight voltage rise and drop. Stuck-at, pulse output, and internal oscillation faults can be caused by this bridging structure.

4. If the both ends of a pair of cascaded transmission gates have opposite logic values and they are turned on (Fig. 4.3(b)), then both ends keep their logic values. However, the voltage of the net between both transmission gates becomes 0.561V (Weak-0). This bridge mainly brings about stuck-at faults.

5. When the outputs of two inverters are directly connected (Fig. 4.4), the voltage of the shorted net is 0.997V (Weak-1). It can cause pulse output faults.

Further, Section 4 shows how to apply these basic bridging characteristics to analyze fault behaviors.

### 4.2 Analysis of fault behaviors

In this section it is demonstrated that bridge defects in the synchronizer can manifest themselves as stuck-at faults, functional timing faults, pulse output faults, internal oscillation faults, or undefined output faults. Also, we present examples to explain each fault type. Initially, we inject bridging defects without resistance to maximize the fault effect (Sections 4.2.1 - 4.2.5). This enables us to find most of the fault types caused by bridging defects. Then, we inject different resistances to each bridging defect to find more faults (Sections 4.2.6 - 4.2.7). In the following analysis, we assume that the synchronizer receives its input from an inverter (instead of $V_{DD}$ or the ground) to simulate the real situation.

#### 4.2.1 Stuck-at fault (SAF)

Bridging defects can cause stuck-at-0 (SAF0) or stuck-at-1 (SAF1) fault effect at the synchronizer output. We present three different cases which give rise to this fault type.
Signal conflict for one transmission gate

Fig. 4.5 shows the synchronizer with $bc26_{DFF1}$ where Fig. 4.5(a) gives the state of the synchronizer initialized with input logic-0. While the input keeps logic-0, there is no signal conflict as shown in Fig. 4.5(a). When the input is changed to logic-1 and CLK is low, there is a signal conflict at the transmission gate (t1) between nodes 1 and 2 of DFF1 as shown in Fig. 4.5(b). As explained in Section 4.1.3, this condition makes the transmission gate work in the saturation region and show a large resistance effect. Even though both ports of the transmission gate show voltage drop and rise, they can keep their logic values (Fig. 4.3(a)). As a result, no input change can pass through DFF1 and the defect behaves as a stuck-at-0 fault (SAF0). In addition, if the synchronizer with $bc26_{DFF1}$ is initialized by input logic-1, on the contrary, input logic-0 cannot pass through DFF1 and the defect behaves as a stuck-at-1 fault (SAF1). Thus, defect $bc26_{DFF1}$ occurred to the synchronizer behaves as a SAF0 or SAF1 depending on the initial state of the synchronizer.

![Figure 4.5: Signal conflict for a transmission gate in synchronizer with $bc26_{DFF1}$](image)

(a) Input = logic-0

(b) Input = logic-1 and CLKB = 1

Signal conflict for two transmission gates

Fig. 4.6 shows the synchronizer with bridging defect $bb22$, and the initial state is given in Fig. 4.6(a). When the input is logic-0, there is no signal conflict as shown in Fig. 4.6(a). When the input is changed to logic-1 and CLK is low, a shorted net (the dashed line in Fig. 4.6(b)) is created through two transmission gates, $t_1$ and $t_5$, (similar to the case shown in Fig. 4.3(b)). HSPICE simulation demonstrates that the voltage at the node 2 in DFF2 under $bb22$ is about 0.561V (Fig. 4.3(b)), which cannot switch the output behavior of
the next inverter \((i_5)\) according to Fig. 3.2. Therefore, the input change cannot pass through DFF2 and the synchronizer with \(bb22\) behaves as a stuck-at-0 fault (SAF0).

![Diagram](image)

(a) Input = logic-0

(b) Input = logic-1 and CLKB = 1

Figure 4.6: Signal conflict for two transmission gates in synchronizer with \(bb22\)

**Bridging between input and output of an inverter**

Bridging defect \(bc23_{DFF1}\) creates a short between the input and output of inverter \(i_1\) shown in Fig. 3.1. As explained in Sec. 4.1.1, when CLK is high, the voltage of the short becomes \(V_{LT H}\) regardless of the input value applied to the synchronizer as shown in Fig. 4.1(b). Then, the voltage \(V_{LT H}\) is propagated to the input of inverter \(i_3\). The inverter \(i_3\) is driving its output towards \(V_{LT H}\) (according to Fig. 3.2) while CLK is high, and the output of \(i_3\) reaches \(V_{LT H}\) if CLK is held high for sufficient time (10.6\(ns\)). In our experiment, CLK changes to low before the output of \(i_3\) reduces to \(V_{LT H}\). We have observed that the output of \(i_3\) is about 1.2V (which is identified as logic-1) when CLK switches to low. However, when CLK is low, inverter \(i_3\) restores to its previous state by enabling the feedback loop. As a result, the synchronizer behaves like a stuck-at-1 fault (SAF1).

**4.2.2 Functional timing fault**

If a bridging defect generates a shorted net to bypass a portion of the circuit, the synchronizer can generate an earlier-than-expected output, either inverted or non-inverted. In addition, if a bridging defect creates a signal conflict in the synchronizer, the synchronizer can generates an inverted output transition with or without extra delay.
Even though the synchronizer with those faults works functionally, an earlier-than-expected output fault can reduce the mean time to failure (MTTF) [19] of the synchronizer, and a later-than-expected output fault can lead to system performance degradation.

**Early transition (TFT1) and inverted early transition (TFT2)**

Defect $bc_{12_{DFF1}}$ connecting nodes 1 and 2 of DFF1 as shown in Fig. 4.7(a) is given as an example of a bridging defect bypassing a transmission gate. If an input change is applied when CLK is high, the input change can propagate to the second latch of DFF1 directly. Then, the synchronizer will generate a new output one clock cycle earlier than expected. However, if the input change is applied when CLK is low, the synchronizer will operate normally. Defect $bc_{13_{DFF1}}$ connecting nodes 1 and 3 as shown in Fig. 4.7(b) is given as an additional example to show an inverted output earlier than expected. This defect causes the input of the synchronizer to propagate to the second latch of DFF1 directly, if the input change is applied when CLK is high. Then, the DFF2 latches the inverted output of DFF1 and the synchronizer generates an inverted output one clock cycle earlier than expected. Consequently, depending on how many latches are bypassed by this type of bridging defects, the synchronizer may generate an earlier-than-expected output, either inverted or non-inverted.

![Figure 4.7: TFT1 and TFT2](image-url)
Inverted output fault (IDO)

Bypassing a portion of circuits in the synchronizer as discussed before can cause an early transition or inverted early transition at the output. However, depending on when an input signal is applied to the synchronizer, the output of the synchronizer can also generate an inverted output without early transition. For example, consider the synchronizer with $bc_{13DFF1}$. Because of the defect, the first latch of DFF1 in the synchronizer turns into the circuits as shown in Fig. 4.2. If the input of the synchronizer is changed to logic-1 (from logic-0) when CLK is low, nodes 1, 2, and 3 of DFF1 (Fig. 3.1) have voltage 1.387V (Fig. 4.2(c)). When CLK is high, node 5 is set to logic-1 and node 6 is set to logic-0 in DFF1. This logic-0 will propagate to the synchronizer output with one more clock cycle ($CLKB = 1$ and then $CLK = 1$). Thus, the synchronizer output is inverted, although the timing is correct.

Inverted late transition (TFT4)

Fig. 4.8 shows the synchronizer with $bc_{36DFF1}$. Figs. 4.8(a)-(b) gives the internal state of the synchronizer when its input is logic-0. Due to the injected bridging defect, nodes 3, 5, and 6 get 1.387V (as described in Fig. 4.2) which is the weak logic-1 as shown in Fig. 4.8(b). When the input of the synchronizer is set to logic-1 in $CLKB = 1$, there is a signal conflict along the shorted net created by the bridging defect in Fig. 4.8(c). As shown, the second latch of DFF1 quickly forms a signal loop as node 6 originally has weak logic-1 (1.387V). However, simultaneously, node 3 of DFF1 is trying to make a 1 to 0 transition. This 1 to 0 transition will bypass the transmission gate between nodes 3 and 5 of DFF1. Thus, the synchronizer output is inverted due to the bridging effect. However, its inverted output might occur with or without one clock delay, depending on how late the input of the synchronizer is changed.

Fig. 4.9 shows the HSPICE simulation waveform of the synchronizer with $bc_{36DFF1}$. This experiment was performed by applying input test patterns at several different time locations in one clock cycle of 2ns. Variable $t_{dist}$ stands for a time distance from the rising edge of the clock signal to the application of an input pattern. If $t_{dist}$ is negative (positive), then an input pattern is applied before (after) the rising edge of the clock. We assume that 0V to 0.3 $V_{DD}$ is logic-0 and 0.7 $V_{DD}$ to $V_{DD}$ is logic-1, respectively.

The input signal is applied to the synchronizer at two different time points, $t_{dist} = -0.9ns$ and $-0.7ns$ where $-0.9ns$ is earlier than $-0.7ns$. As shown in Fig. 4.9, if the input signal is applied at $t_{dist} = -0.9ns$ (dotted line), the fault behavior of the synchronizer is IDO. However, if the input signal is applied at $t_{dist} = -0.7ns$ (solid line), the synchronizer shows the inverted output one clock cycle later than expected (TFT4). Note that the clock signal in Fig. 4.9 is represented by dashed lines.
Figure 4.8: Internal signal transitions of synchronizer with $bc36_{DF1}$ for logic-0 to 1 input change for different clock cycles
Figure 4.9: Waveforms of synchronizer with $bc\text{36}_{\text{DF1}}$ for logic-0 to 1 input change at $t_{\text{dist}} = -0.9\text{ns}$ (dotted line) and $t_{\text{dist}} = -0.7\text{ns}$ (solid line)
The different fault behaviors result from that the injected bridging defect induces extra delay for a signal transition on the shorted net, and it leads to a signal race between that signal transition on the shorted net and the clock signal at the first transmission gate of DFF2. If a logic-0 to 1 input signal transition occurs late (e.g., at $t_{\text{dist}} = -0.7\,\text{ns}$), the resulting logic-1 to 0 signal transition on the shorted net cannot be across the $V_{LT H}$ while $\text{CLKB} = 1$. When $\text{CLK} = 1$, its voltage bounces back to about 1.387V. But, in the upcoming $\text{CLKB} = 1$, it exercises the full cycle of logic-1 to 0 transition and finally finishes the transition. Note that the fault behavior of a bridging defect can change for different values of $t_{\text{dist}}$, and this indicates that the defect behavior depends on the clock frequency used during testing.

Besides, there is one more timing fault, delayed output (DOF) fault, which is not discussed in detail. This fault gives an extra delay which is less than one clock cycle because of the resistance created by a bridging defect.

### 4.2.3 Pulse output fault (POF)

A bridging defect may cause the synchronizer to generate pulses. There are two types of pulses generated by the synchronizer with a bridging defect: input-dependent and input-independent. If pulses generated are dependent on a specific input logic value of the synchronizer, the synchronizer creates pulses with different frequency or different phase based on the input logic value. However, if the input cannot propagate to the synchronizer, then the synchronizer may generate pulses which are input-independent.

**Input-dependent pulses (POF1)**

Fig. 4.10 shows the synchronizer with $bc17_{\text{DFF2}}$ for the input with logic-0. Fig. 4.10(a) shows the initial state of the synchronizer. When CLK is high, there is a signal conflict along the shorted net created by $bc17_{\text{DFF2}}$ and the voltage becomes 0.997V (Fig. 4.10(b)). This bridging effect has been discussed in Fig. 4.4. When CLK is low in Fig. 4.10(c), logic-0 of the node 7 in DFF1 updates the node 5 in DFF1 and the output of DFF1 becomes logic-1. The output of DFF1 is then propagated to the first latch of DFF2. When CLK is high in Fig. 4.10(d), the first latch of DFF1 updates the second latch of DFF1, and the first latch of DFF2 updates the second latch of DFF2. There is no signal conflict on the shorted net in this case. When CLK is low in Fig. 4.10(e), the synchronizer has the same state as Fig. 4.10(a) and the analysis can be repeated. The synchronizer with $bc17_{\text{DFF2}}$ shows $f_{\text{CLK}}/2$ pulses at the output when the input is logic-0.

Fig. 4.11 gives the case of the synchronizer with $bc17_{\text{DFF2}}$ for the input of logic-1. In fact, when the input is set to logic-1 from logic-0, the initial state of the synchronizer can be either Fig. 4.10(a) or Fig. 4.10(c). In Fig. 4.11(a), the input of the synchronizer is changed to logic-1 at the state in Fig. 4.10(a). The
Figure 4.10: Internal state transitions of synchronizer with bc17$_{\text{DFF2}}$ when input = logic-0 for different clock cycles
synchronizer changes states as shown in Fig. 4.11(b) to Fig. 4.11(e), and then repeats between Fig. 4.11(d) and Fig. 4.11(e). So, the synchronizer with \( bc_{17,DF2} \) for the input of logic-1 gives pulses with \( f_{CLK} \) at the output. The input of the synchronizer can also be changed from logic-0 to logic-1 when the state of the synchronizer is Fig. 4.10(c). The analysis of the synchronizer state change begins from Fig. 4.11(c).

**Input-independent pulses (POF2)**

A bridging defect can cause the synchronizer to generate input-independent pulses at its output. When the synchronizer generates input-independent pulses, its input cannot propagate to the synchronizer due to the bridging defect. Fig. 4.12 shows the synchronizer with \( bb_{23} \). The \( bb_{23} \) bridging defect brings about a loop with an odd number of inverters consisting of \( i_1 \) and \( i_3 \) of DFF1 and \( i_5 \) of DFF2. Thus, whenever CLK changes its logic value, the inverters along the loop also changes their logic values, which are observed at the output of the synchronizer.

Fig. 4.12(a) gives the initial state of the synchronizer when CLK is high. When CLK is low, the output of DFF1 updates the first latch of DFF2 and the output of the first latch of DFF2 also updates the first latch of DFF1 due to the shorted net created by \( bb_{23} \) as shown in Fig. 4.12(b). Since nodes 1 and 2 of the transmission gate in DFF1 have opposite voltages, the transmission gate works in the saturation mode as explained in Fig. 4.3(b). Thus, the input applied to the synchronizer cannot be latched to the first latch of DFF1. When CLK is high in Fig. 4.12(c), the first latch of DFF1 updates the second latch of DFF1, and the first latch of DFF2 updates the second latch of DFF2, respectively. The output of the synchronizer is set to logic-0. Fig. 4.12(d) shows the internal state of the synchronizer in upcoming \( CLK_B = 1 \), where the state of the internal nodes are opposite to the case of Fig. 4.12(b). As a result, the synchronizer with \( bb_{23} \) gives pulses with the frequency of \( f_{CLK}/2 \) at its output.

This fault behavior results from that the transmission gate between nodes 1 and 2 of DFF1 blocks the propagation of the input of the synchronizer, and the loop with an odd number of inverters created by the bridging defect makes a signal oscillation by switching the clock.

**4.2.4 Internal oscillation fault (IOF)**

Bridging defects can introduce a loop with an odd number of inverters when CLK is low or high, and signal oscillation along the loop occurs. The fault behavior of a synchronizer with such bridging defects depends on the frequency of the signal oscillation generated on the loop, and the clock frequency which is used to fetch the signal into the latch.

Bridging defect \( bc_{27,DF1} \) introduces a loop with odd number of inverters consisting of \( i_1 \), \( i_3 \), and \( i_4 \) of
Figure 4.11: Internal signal transitions of synchronizer with $bc_{17}^{DF2}$ when input = logic-1 for different clock cycles
Figure 4.12: Internal signal transitions of synchronizer with $bb23$ when input = logic-0 for different clock cycles
DFF1 when $CLK = 1$ as shown in Fig. 4.13(a), and signal oscillation occurs along the loop. The logic value to be latched into the first latch of DFF2 is determined by the logic value of signal line 6 of DFF1 when CLK becomes low. Fig. 4.13(b) and Fig. 4.13(c) show the cases in which the logic value of signal line 6 of DFF1 is logic-1 and logic-0, respectively. The logic value of signal line 6 of DFF1 is latched into the first latch of DFF2 when CLK is low. When CLK is high, the latched logic value in the first latch of DFF2 shows at the output of the synchronizer. Especially, in Fig. 4.13(c), signal lines 1 and 2 in DFF1 have different logic values and therefore, both transistors in the transmission gate between signal line 1 and signal line 2 operate in the saturation region. This fact disables any input change for the synchronizer to propagate to the first latch of DFF1.

![Fig. 4.13: Two different outputs of synchronizer with $bc27_{DFF1}$ when input = logic-0 and $freq_{CLK} = 500MHz$](image)

Fig. 4.14 shows the HSPICE simulation result of the synchronizer with $bc27_{DFF1}$ and $f_{CLK} = 500MHz$. When CLK is high, there is a loop created and signal oscillation occurs along the loop. When CLK becomes low, the output of the second latch of DFF1 (signal line 6) can be either logic-0 or logic-1. In this case, the synchronizer with $bc27_{DFF1}$ shows pulses with $f_{CLK}/2$ in its output regardless of its input logic value.
Fig. 4.15 shows the HSPICE simulation result of the synchronizer with $bc_{27DFF1}$ and $f_{CLK} = 250$MHz. In this case, the output of the second latch of DFF1 always has logic-1. Thus, the $bc_{27DFF1}$ defect causes the synchronizer to exhibit stuck-at-1 fault behavior when $f_{CLK} = 250$MHz.

Figure 4.14: Waveforms of synchronizer with $bc_{27DFF1}$ when $f_{CLK} = 500$MHz for input signal and different nodes (dotted lines)

4.2.5 Undefined output fault (UOF)

In contrast to the example in Fig. 3.2, where it is known that the logic threshold voltage of the input of an inverter is 0.874V for the considered technology and 1.8V $V_{DD}$, the output of the synchronizer can drive any type of gate, where each gate type corresponds to different values of logic threshold voltage within the range $0.3 \cdot V_{DD}$ to $0.7 \cdot V_{DD}$. Without further knowledge of the gate type that is driven by the synchronizer, the logic outcome of an output voltage in the range $0.3 \cdot V_{DD}$ to $0.7 \cdot V_{DD}$ cannot be known, and therefore in such cases the output is considered to be undefined in terms of logic value. Assume the synchronizer has
Figure 4.15: Waveforms of synchronizer with $bc27_{DF1}$ when $f_{CLK} = 250$MHz for input signal and different nodes (dotted lines)
(a) When CLKB = high

(b) When CLK = high

Figure 4.16: Synchronizer with bc56_{DFF2}

defect bc56_{DFF2} as shown in Fig. 4.16. The bridging defect makes a direct connection between the input and output (nodes 5 and 6 of DFF2) of the inverter in the second latch of DFF2. Our HSPICE simulation demonstrates that as explained in Fig. 4.1, the output of the synchronizer shows a voltage $V_{LTH}$ when CLKB is high. However, the output of the synchronizer is 0.987V (0.644V) when CLK is high and the input is logic 0 (logic 1). The output voltages of the synchronizer for all cases are in the range of undefined voltage level.

4.2.6 One-time pulse fault (OTP)

In the above discussion (Sections 4.2.1 - 4.2.5), we assumed zero resistance for all bridging defects. Starting from this subsection, we discuss more fault effects caused by bridging defects with non-zero resistances.

A bridging defect with non-zero resistance may cause the synchronizer to generate an one-time pulse at its output, when the input is changed. The synchronizer with defect $bb13$ at 1.07kΩ is given as an example in Fig. 4.17. For the case of 0Ω resistance by $bb13$, the fault behavior of the synchronizer is TFT2. That is, the synchronizer gives an opposite signal transition one clock cycle earlier as the response to its input signal change. However, the synchronizer with $bb13$ whose resistance is 1.07kΩ shows an one-time pulse as its fault behavior.

Fig. 4.17(a) gives the initial state of the synchronizer to be analyzed. When CLK is high, the voltage of the signal net 1 of DFF1 is 0.448V and that of the signal net 3 of DFF2 is 0.93V, respectively. The voltage of 0.93V (weak logic-1) is propagated to the second latch of DFF2. Unfortunately, the time period of $CLK = 1$
is not long enough to finish the $V_{DD}$ to 0V signal transition at the output of the second latch of DFF2 as its response to 0.93V. When CLK is low, the output voltage of DFF2 bounces back to $V_{DD}$ as shown in Fig. 4.18. The failure of changing the second latch state in DFF2 by the first latch comes from the degraded voltage (0.93V) which should be 1.8V at node 3 of DFF2.

If the input is set to logic-1 from logic-0 when CLK is low, the voltage of the signal net 3 of DFF2 becomes 1.8V because both edges of the shorted net have the same voltage level, 1.8V (Fig. 4.17(b)). When CLK is high, the synchronizer gives logic-0 for its output (Fig. 4.17(c)). However, in the next coming $CLKB = 1$, there is a signal conflict on the shorted net. The voltage of the signal net 1 of DFF1 becomes 1.13V and that of the signal line 3 of DFF2 becomes 0.589V, respectively (Fig. 4.17(d)). So, the output of the synchronizer is back to logic-1 when $CLK = 1$ (Fig. 4.17(e)). From this on, the output of the synchronizer keeps logic-1 regardless of the CLK value (Fig. 4.18). Consequently, defect $bb13$ with 1.07k$\Omega$ causes the synchronizer to generate a 1-0-1 one-time pulse at its output if a logic-0 to 1 input change is applied.

### 4.2.7 Late transition fault (TFT3)

Fig. 4.19(a) shows the initial state of the synchronizer with $bc37_{DFF1}$ at 747 $\Omega$ when its input is logic-0. Fig. 4.19(b) gives the circuit diagram of DFF1 in Fig. 4.19(a) when CLK is low. At this time, both voltage levels of nodes 3 and 7 are 1.8 V. When the input of the synchronizer is changed to logic-1 and CLK is low, signal conflict takes place on the shorted net as shown in Fig. 4.19(c). Fig. 4.19(d) gives the circuit diagram of DFF1 in Fig. 4.19(c). When the input of the synchronizer is changed to logic-1 and the clock is low, the NMOS of inverter $i_1$ is turned on and begins to discharge the input of inverter $i_2$. However, at the same time, the PMOS of inverter $i_4$ is also turned on and there is a current path from $V_{DD}$ to the ground (dotted line in Fig. 4.19(c) and Fig. 4.19(d)). There is a voltage difference between nodes 3 and 7 because of the resistance located between them (see time = 9ns, Fig. 4.21). This resistance makes the voltage of node 3 keep lower than that of node 7.

Fig. 4.20(a) shows the circuit diagram in the beginning of $CLK = 1$ following the above discussion in Fig. 4.19(c). At this time, the input voltage of inverter $i_2$ is 0.801 V which is not low enough to turn on the PMOS of $i_2$ fully and the input of inverter $i_1$ has weak logic-1. However, the PMOS of inverter $i_4$ is fully turned on and it is still strongly charging the inputs of inverters $i_2$ and $i_3$. As a result, the charging current of $i_4$ is stronger than the discharging current of $i_1$. Thus, as shown in Fig. 4.21 (t = 10ns) the voltage of nodes 3 and 7 in Fig. 4.20(d) bounces back to $V_{DD}$ when CLK = 1. As a result, the input of i1 cannot remain in logic-1 and turns into logic-0 instead. That is, the input logic-1 just passing the transmission gate between nodes 1 and 2 of DFF1 fails to be maintained in the first latch of DFF1.
Figure 4.17: Internal state changes of synchronizer with $bb_{13}$ at $R = 1.07k\Omega$ for logic-0 to 1 input change and different clock cycles
Figure 4.18: Waveform of the synchronizer with $bb_{13}$ when $R = 1.07k\Omega$ for input and different nodes (dotted lines)
Figure 4.19: Internal states of synchronizer with \(bc37_{DF\text{F}_1}\) at 747 Ω for different input logic values
Figure 4.20: Voltage variation of internal nodes of synchronizer with $bc37_{DFF1}$ at $747 \, \Omega$ for two consecutive $CLK = 1$
As shown in Fig. 4.21, when CLKB = 1 again and the input is still logic-1, node 3 has a full cycle to drop its voltage which can be as low as 0.669V. When CLK = 1 again as shown in Fig. 4.20(b) and Fig. 4.21 (t = 11ns), the voltage of node 3 is low enough to drive the input of i1 to logic-1. As a result, logic-1 applied to the synchronizer input has been locked into the first latch of DFF1 and this makes the synchronizer take an extra cycle to present its response to the output. Mainly, this one clock cycle delay comes from the late arrival time of input signal change shown in the HSPICE waveform in Fig. 4.21. This creates a signal race between the signal transition on the shorted net and CLK at time 9ns of Fig. 4.21. Fig. 4.22 shows the detailed HSPICE simulation waveforms of the synchronizer with $bc37_{DFF1}$ at 747 Ω.

![Signals at nodes 3 (dotted line) and 7 (solid line) in DFF1](image)

Figure 4.21: Nodes 3 (dotted line) and 7 (solid line) of the synchronizer with $bc37_{DFF1}$ at 747 Ω
Figure 4.22: Waveform of the synchronizer with $bc37_{DFF1}$ at $747 \, \Omega$ for input signal and different nodes (dotted lines)
4.3 Summary of observed fault behaviors

4.3.1 Fault behavior observation for 180nm technology model

We have performed an experiment on the synchronizer implemented using the 180nm TSMC technology model mentioned in Section 3.2 by considering all possible bridging defects to check their possible fault behaviors. Table 4.1 shows the categories of all fault behaviors obtained by this experiment for the synchronizer. The first column in Table 4.1 gives the categories of all fault behaviors. The second column shows more detailed fault types belonging to each category, and the detailed fault behavior of each type is described in the third column. Each of the fault categories has been discussed in the previous section.

Table 4.1: Category of all bridging faults

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-At Fault</td>
<td>SAF0</td>
<td>Stuck-at-0</td>
</tr>
<tr>
<td></td>
<td>SAF1</td>
<td>Stuck-at-1</td>
</tr>
<tr>
<td>Functional Timing Fault</td>
<td>TFT1</td>
<td>The output of the synchronizer is observed earlier than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT2</td>
<td>The inverted output of the synchronizer is observed earlier than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT3</td>
<td>The output of the synchronizer is observed at least one clock cycle later than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT4</td>
<td>The inverted output of the synchronizer is observed at least one clock cycle later than expected.</td>
</tr>
<tr>
<td></td>
<td>DOF</td>
<td>The output with delay of the synchronizer is observed.</td>
</tr>
<tr>
<td></td>
<td>IDOF</td>
<td>The inverted output with delay of the synchronizer is observed.</td>
</tr>
<tr>
<td>Pulse Output Fault</td>
<td>POF1</td>
<td>The synchronizer generates a pulse with regard to its input signal.</td>
</tr>
<tr>
<td></td>
<td>POF2</td>
<td>The synchronizer generates a pulse regardless of its input signal.</td>
</tr>
<tr>
<td>One-Time Pulse Fault</td>
<td>OTP0</td>
<td>The output of the synchronizer shows 1-0-1 one-time pulse.</td>
</tr>
<tr>
<td></td>
<td>OTP1</td>
<td>The output of the synchronizer shows 0-1-0 one-time pulse.</td>
</tr>
<tr>
<td>Internal Oscillation Fault</td>
<td>IOF</td>
<td>A signal oscillation occurs in the synchronizer by a loop, and the output of the synchronizer depends on the applied clock frequency.</td>
</tr>
<tr>
<td>Undefined Output Fault</td>
<td>UOF</td>
<td>The output voltage of the faulty synchronizer is in the range of ( V_{DD} \cdot 0.3 ) to ( V_{DD} \cdot 0.7 ).</td>
</tr>
</tbody>
</table>

Tables 4.2-4.4 show the synchronizer fault behaviors with each bridging defect depending on the fault location. The first column identifies a bridging fault. The second column is the input patterns applied to the synchronizer. The third column shows the fault behavior of the synchronizer with a bridging fault which has no resistance. The fourth column presents the critical resistance - the minimum resistance from which the synchronizer with a bridging fault behaves like a fault-free synchronizer. Based on the critical resistance of each fault, we have performed more experiments on the synchronizer with three different resistances for
the fault. These three resistances are evenly distributed from 0Ω to the critical resistance. For example, if the critical resistance is 1kΩ, R1, R2, and R3 are 250, 500, and 750Ω, respectively. So, the fifth, sixth and seventh columns show the fault behaviors of the synchronizer with the three different resistances for the fault. There is a new fault, DOF, which is not discussed before. This fault gives an extra delay which is less than one clock cycle, because of the resistance created by a bridging defect.

Table 4.2 shows the fault behaviors for each bridging defect located in the first D flip-flop (DFF1) of the synchronizer. The synchronizer shows different fault behaviors depending on the signal arrival time when an input signal is applied. For example, in the case of bc12_DFF1, if the input signal is applied when \( t_{\text{dist}} \) is negative, it works like a fault-free synchronizer. However, if the input signal is applied when \( t_{\text{dist}} \) is positive, the output will generate a signal transition one clock cycle earlier than expected (TFT1). Moreover, the synchronizer shows different fault behaviors for different input signal patterns. For example, when a logic-0 to 1 input signal transition is applied to the synchronizer with bc14_DFF1, the synchronizer presents an output signal transition one clock cycle earlier than expected (TFT1). However, when a logic-1 to 0 input signal transition is applied, it works like a fault-free synchronizer. Different resistance of a bridging defect can also result in different fault behaviors. For example, the synchronizer with bc15_DFF1 at resistance R1 gives TFT2 as its fault behavior when the input transition is from logic-1 to 0. However, the same fault at resistance values R2 and R3 causes stuck-at-1 faults (SAF1). For bc27_DFF1 with non-zero resistance, if the input is logic-0, the synchronizer behaves as stuck-at-1 fault (SAF1). However, if the input is logic-1, it generates unexpected pulses, and the width of each pulse is getting wider by increasing the injected resistance. We extended the simulation time to search the critical resistance, however pulses are observed. Thus, we concluded that the critical resistance is not available in this experiment.

Table 4.3 shows the fault behaviors for each bridging defect located in the second D flip-flop (DFF2) of the synchronizer. Even though a bridging defect is located at the same place in both D flip-flops, the fault behavior can be different. For example, the synchronizer with bc56_DFF2 shows different fault behavior (UOF) from the synchronizer with bc56_DFF1 whose fault behavior is SAF0. Moreover, the synchronizer with a bridging defect shows different fault behaviors based on the applied input signal pattern and the resistance of the bridging defect. Note that input arrival time is not a critical factor for the bridging faults in DFF2. For bc27_DFF2, the critical resistance is not available with the same reason as bc27_DFF1.

Table 4.4 shows fault behaviors of the synchronizer if a bridging defect is located between two D flip-flops. Similar to the previous experimental results, the fault behavior of the synchronizer depends on the fault location, the input signal pattern, its arrival time, and the resistance of a bridging defect. For bb23 and bb27, the synchronizer generates pulses, and the width of each pulse is getting wider by increasing the
Table 4.2: Fault behaviors with bridging defects in DFF1

<table>
<thead>
<tr>
<th>Fault name</th>
<th>Input pattern</th>
<th>0Ω</th>
<th>Critical resistance</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>b_{c12}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT1^{1}</td>
<td>2.9kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1^{1}</td>
<td>4.4kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>b_{c13}^{DIFF}</td>
<td>0 → 1</td>
<td>IDOF/TFT2^{2}</td>
<td>1.9kΩ</td>
<td>IDOF</td>
<td>IDOF</td>
<td>IDOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>IDOF/TFT2^{2}</td>
<td>1.7kΩ</td>
<td>IDOF</td>
<td>IDOF</td>
<td>IDOF</td>
</tr>
<tr>
<td>b_{c14}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT1^{1}</td>
<td>254Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>b_{c15}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>11.6kΩ</td>
<td>IDOF</td>
<td>SAF1</td>
<td>SAF1^{2}</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>11.6kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c16}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT1^{1}</td>
<td>527Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1^{1}</td>
<td>253Ω</td>
<td>Fault Free(F/F)^{3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b_{c17}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT2^{2}</td>
<td>997Ω</td>
<td>TFT2</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2^{2}</td>
<td>980Ω</td>
<td>TFT2</td>
<td>OTP1</td>
<td>OTP1</td>
</tr>
<tr>
<td>b_{c18}^{DIFF}</td>
<td>both</td>
<td>SAF1</td>
<td>3.9kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c19}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>6.7kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>6.3kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td></td>
</tr>
<tr>
<td>b_{c20}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>10.4kΩ</td>
<td>SAF1</td>
<td>SAF1/OTP0^{7}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>10.4kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c21}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>9kΩ</td>
<td>SAF0</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>6.6kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c22}^{DIFF}</td>
<td>both</td>
<td>IOF</td>
<td>Not Available(N/A)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b_{c23}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>1.8kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c24}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT1^{3}</td>
<td>1.2kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1^{3}</td>
<td>1.4kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td></td>
</tr>
<tr>
<td>b_{c25}^{DIFF}</td>
<td>0 → 1</td>
<td>IDOF/TFT4^{4}</td>
<td>1.9kΩ</td>
<td>IDOF</td>
<td>IDOF</td>
<td>IDOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>IDOF/TFT4^{4}</td>
<td>1.8kΩ</td>
<td>IDOF</td>
<td>IDOF</td>
<td>IDOF</td>
</tr>
<tr>
<td>b_{c26}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>996Ω</td>
<td>SAF0</td>
<td>SAF0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>253Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td></td>
</tr>
<tr>
<td>b_{c27}^{DIFF}</td>
<td>both</td>
<td>TFT2^{5}</td>
<td>11.6kΩ</td>
<td>IDOF</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c28}^{DIFF}</td>
<td>0 → 1</td>
<td>TFT1^{6}</td>
<td>37Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1^{6}</td>
<td>73Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td></td>
</tr>
<tr>
<td>b_{c29}^{DIFF}</td>
<td>0 → 1</td>
<td>IDOF/TFT2^{7}</td>
<td>878Ω</td>
<td>TFT2</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>IDOF/TFT2^{7}</td>
<td>68Ω</td>
<td>TFT2</td>
<td>TFT2</td>
<td>POF1</td>
</tr>
<tr>
<td>b_{c30}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>4kΩ</td>
<td>SAF0</td>
<td>F/F^{16}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>4kΩ</td>
<td>SAF0</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c31}^{DIFF}</td>
<td>both</td>
<td>SAF1</td>
<td>6.7kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td></td>
</tr>
<tr>
<td>b_{c32}^{DIFF}</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>1.7kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>1.5kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td></td>
</tr>
</tbody>
</table>

1. TFT1 in \( t_{dist} = -0.1\text{ns} \sim 0.3\text{ns} \). 2. TFT2 in \( t_{dist} = -0.1\text{ns} \sim 0.3\text{ns} \), and IDOF otherwise. 3. TFT1 in \( t_{dist} = -0.1\text{ns} \sim 0.3\text{ns} \). 4. Except \( t_{dist} = -0.5\text{ns} \sim -0.3\text{ns} \). 5. Except \( t_{dist} = -0.3\text{ns} \). 6. TFT4 in \( t_{dist} = -0.7\text{ns} \sim -0.3\text{ns} \), and IDOF otherwise. 7. IDOF in \( t_{dist} = -0.7\text{ns} \sim -0.3\text{ns} \), and TFT2 otherwise. 8. Except \( t_{dist} = -0.7\text{ns} \sim -0.3\text{ns} \). 9. Except \( t_{dist} = -0.9\text{ns} \sim -0.3\text{ns} \). 10. IDOF in \( t_{dist} = -0.7\text{ns} \sim -0.3\text{ns} \), and TFT2 otherwise. 11. IDOF in \( t_{dist} = -0.5\text{ns} \sim -0.3\text{ns} \), and TFT2 otherwise. 12. OTP0 if \( 11.5kΩ \leq R \leq 11.6kΩ \). 13. TFT3 at \( R = 244Ω \). 14. SAF1 if \( t_{dist} = -0.5\text{ns} \) and OTP0 if \( t_{dist} = 0.5\text{ns} \). 15. TFT3 if \( R = 10.4kΩ \). 16. OTP0 if \( R = 4kΩ \).
Table 4.3: Fault behaviors with bridging defects in DFF2

<table>
<thead>
<tr>
<th>Fault name</th>
<th>Input pattern</th>
<th>0Ω</th>
<th>Critical resistance</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bc12DFF2</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>3.4kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>6.1kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bc13DFF2</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>2.9kΩ</td>
<td>DOF</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>IDOF</td>
<td>7.3kΩ</td>
<td>TFT1</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td>bc14DFF2</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>273Ω</td>
<td>TFT1</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>459Ω</td>
<td>SAF1</td>
<td>DOF</td>
<td>IDOF</td>
</tr>
<tr>
<td>bc15DFF2</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>10.6kΩ</td>
<td>TFT2</td>
<td>OTP0</td>
<td>OTP0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>20.1kΩ</td>
<td>SAF1</td>
<td>POF1</td>
<td>DOF</td>
</tr>
<tr>
<td>bc16DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>1.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>991Ω</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bc17DFF2</td>
<td>both</td>
<td>POF1</td>
<td>Not Available(N/A)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bc23DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>5.5kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>5.3kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>DOF</td>
</tr>
<tr>
<td>bc24DFF2</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>6.4kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>6.4kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td>bc25DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>9.3kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>19kΩ</td>
<td>SAF1</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td>bc26DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>8.1kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>6.6kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bc27DFF2</td>
<td>both</td>
<td>IOF</td>
<td>Not Available(N/A)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bc34DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>3.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>3.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>DOF</td>
</tr>
<tr>
<td>bc35DFF2</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>8.1kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>5.3kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bc36DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>2.9kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>2.6kΩ</td>
<td>TFT2</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bc37DFF2</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>620Ω</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>661Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bc45DFF2</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>10.6kΩ</td>
<td>TFT2</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>20.1kΩ</td>
<td>SAF1</td>
<td>POF1</td>
<td>DOF</td>
</tr>
<tr>
<td>bc46DFF2</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>527kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>270kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bc47DFF2</td>
<td>0 → 1</td>
<td>POF1</td>
<td>910Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>878Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td>bc56DFF2</td>
<td>0 → 1</td>
<td>UOF</td>
<td>3.4kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>UOF</td>
<td>5kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bc57DFF2</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>7.3kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>10kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>DOF</td>
</tr>
<tr>
<td>bc67DFF2</td>
<td>0 → 1</td>
<td>UOF</td>
<td>2.8kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>UOF</td>
<td>2.7kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
</tbody>
</table>
injected resistance. However, their critical resistances are not available in this experiment with the same reason as $bc27_{DFF1}$.

Table 4.5 shows the summary of experimental results using the 180nm technology model used depending on the resistance values. The first and fourth columns are fault types observed in the experiment with zero resistance and non-zero resistance, respectively. The second and fifth columns give the number of defects in each fault type and the third and sixth are the percentile of them. Stuck-at fault is around 35% (40%) in the experiment with zero resistance (non-zero resistance). Pulse output fault is around 14% for both experiments. Timing fault type occupies around 43% (34%) in the experiment with zero resistance (non-zero resistance). By introducing non-zero resistance defects, DOF, OTP0, OTP1 and TFT3 are introduced as new fault types with about 9% totally.

Basically, the previous works considered bridging defects occurring in a latch [36, 37, 45]. However, this research is dealing with bridging defects occurring in a synchronizer of two D flip-flops. So, it might not be appropriate to compare this research to the previous works directly. Though the circuits have different structures, stuck-at fault and delay fault presented in this research are quite similar to those in the previous works in fault mechanism. The state modifying behaviors caused by injected bridging defects in [36] are similar to the signal conflict behaviors occurring in the synchronizer. The TFT2 fault identified in this work is similar to the data-feedthrough behavior in [37]. Other fault behaviors are newly discovered in this research.

### 4.3.2 180nm technology model vs. 45nm and 65nm technology models

We conducted further experiments with zero-resistance bridging defects on the synchronizer implemented using the 45nm and 65nm predictive technology models given in [50]. We used the same synchronizer design as that in the 180nm technology model, and applied 1.0V for $V_{DD}$, instead 1.8V. Table 4.6 presents faults which show different fault behaviors from the experiment using the 180nm technology model. The first column identifies bridging defects which show different fault behaviors. The second column is the input patterns applied to the synchronizer. The next three columns show fault behaviors observed using the 180nm, 45nm, and 65nm technology models, respectively. By applying 45nm and 65nm technology models, the behaviors of each inverter are altered, and that fact leads to changing the basic characteristics we introduced in Section 4.1. As a result, five bridging defects show different fault behaviors from those in the experiment with the 180nm technology model. However, all fault behaviors observed are still in the categories of Table 4.1.
Table 4.4: Fault behaviors with bridging defects between DFF1 & DFF2

<table>
<thead>
<tr>
<th>Fault name</th>
<th>Input pattern</th>
<th>0Ω</th>
<th>Critical resistance</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bb12</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>5.2kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>7.1kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb13</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>2.3kΩ</td>
<td>TFT2</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>2.4kΩ</td>
<td>TFT2</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td>bb14</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>60Ω</td>
<td>OTPO</td>
<td>OTPO</td>
<td>OTPO</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>41Ω</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb15</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>10.6kΩ</td>
<td>TFT2</td>
<td>OTPO</td>
<td>OTPO</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>12.9kΩ</td>
<td>TFT2</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb16</td>
<td>0 → 1</td>
<td>UOF</td>
<td>1.1kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>UOF</td>
<td>1.9kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bb17</td>
<td>0 → 1</td>
<td>POF1</td>
<td>1.1Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>1.1Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td>bb22</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>7.3kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>6.2kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb23</td>
<td>both</td>
<td>POF2</td>
<td>Not Available(NA)</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb24</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>9.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>8.5kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb25</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>29.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>9.3kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb26</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>9.2kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>6.7kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb27</td>
<td>both</td>
<td>POF2</td>
<td>Not Available(NA)</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb32</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>8.6kΩ</td>
<td>TFT2</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>7.4kΩ</td>
<td>TFT2</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td>bb33</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>1.2kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>1.2kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td>bb34</td>
<td>0 → 1</td>
<td>POF1</td>
<td>91Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>14Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td>bb35</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>8.5kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>6.8kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb36</td>
<td>0 → 1</td>
<td>POF1</td>
<td>2.6kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>1.3kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bb37</td>
<td>0 → 1</td>
<td>POF1</td>
<td>1.1kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>1.1kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td>bb42</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>5.2kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>7.1kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb43</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>4.3kΩ</td>
<td>DOF</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>7.6Ω</td>
<td>DOF</td>
<td>DOF</td>
<td>DOF</td>
</tr>
<tr>
<td>bb44</td>
<td>both</td>
<td>SAF1</td>
<td>95Ω</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb45</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>10.7kΩ</td>
<td>TFT2</td>
<td>OTPO</td>
<td>OTPO</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>10.8kΩ</td>
<td>TFT2</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb46</td>
<td>0 → 1</td>
<td>POF1</td>
<td>90Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>1.7kΩ</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bb47</td>
<td>0 → 1</td>
<td>POF1</td>
<td>96Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF1</td>
<td>7.9Ω</td>
<td>POF1</td>
<td>POF1</td>
<td>POF1</td>
</tr>
<tr>
<td>bb52</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>6.5kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>5.3kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td>bb53</td>
<td>0 → 1</td>
<td>SAF0</td>
<td>10.7kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF0</td>
<td>10.7kΩ</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td>bb54</td>
<td>0 → 1</td>
<td>ROF</td>
<td>7.9kΩ</td>
<td>ROF</td>
<td>ROF</td>
<td>ROF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>ROF</td>
<td>8kΩ</td>
<td>ROF</td>
<td>ROF</td>
<td>ROF</td>
</tr>
<tr>
<td>bb55</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>10.4kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>10.4kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb56</td>
<td>0 → 1</td>
<td>POF2</td>
<td>8.6kΩ</td>
<td>POF2</td>
<td>POF2</td>
<td>POF2</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>POF2</td>
<td>7.9kΩ</td>
<td>POF2</td>
<td>POF2</td>
<td>POF2</td>
</tr>
<tr>
<td>bb57</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>11.6kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>11.6kΩ</td>
<td>SAF1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb72</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>8.6kΩ</td>
<td>TFT2</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>7.2Ω</td>
<td>TFT2</td>
<td>TFT2</td>
<td>TFT2</td>
</tr>
<tr>
<td>bb73</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>14Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>7.9Ω</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb74</td>
<td>0 → 1</td>
<td>TFT2</td>
<td>87Ω</td>
<td>TFT2</td>
<td>OTPO</td>
<td>OTPO</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT2</td>
<td>90Ω</td>
<td>TFT2</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bb75</td>
<td>0 → 1</td>
<td>TFT1</td>
<td>8.5kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>TFT1</td>
<td>4.5kΩ</td>
<td>TFT1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb76</td>
<td>0 → 1</td>
<td>UOF</td>
<td>2.6Ω</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>UOF</td>
<td>1.3Ω</td>
<td>UOF</td>
<td>UOF</td>
<td>UOF</td>
</tr>
<tr>
<td>bb77</td>
<td>0 → 1</td>
<td>SAF1</td>
<td>997Ω</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 → 0</td>
<td>SAF1</td>
<td>1862Ω</td>
<td>SAF0</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
</tbody>
</table>

1. Except $t_{dist} = -0.3\text{ns}$. 2. Except $t_{dist} = -0.5\text{ns} \sim -0.3\text{ns}$. 3. TFT3 at $R = 1.2\text{kΩ}$. 56
Table 4.5: Summary of experimental results for a 180nm technology model

<table>
<thead>
<tr>
<th>Fault</th>
<th>Zero resistance</th>
<th>Non-zero resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDOF</td>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>IOF</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>POF1</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>POF2</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>SAF0</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>SAF1</td>
<td>39</td>
<td>44</td>
</tr>
<tr>
<td>TFT1</td>
<td>30</td>
<td>14</td>
</tr>
<tr>
<td>TFT2</td>
<td>31</td>
<td>53</td>
</tr>
<tr>
<td>TFT4</td>
<td>1</td>
<td>127</td>
</tr>
<tr>
<td>UOF</td>
<td>8</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 4.6: Experimental results for 45nm and 65nm technology models

<table>
<thead>
<tr>
<th>Fault name</th>
<th>Input pattern</th>
<th>180nm</th>
<th>45nm</th>
<th>65nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>bc14_{DF1}</td>
<td>1 \rightarrow 0</td>
<td>fault-free</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bc36_{DF1}</td>
<td>0 \rightarrow 1</td>
<td>IDOF/TFT4</td>
<td>IDOF/TFT2</td>
<td>IDOF/TFT2</td>
</tr>
<tr>
<td>bc56_{DF1}</td>
<td>both</td>
<td>SAF0</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td>bc16_{DF1}</td>
<td>0 \rightarrow 1</td>
<td>SAF1</td>
<td>SAF0</td>
<td>SAF0</td>
</tr>
<tr>
<td></td>
<td>1 \rightarrow 0</td>
<td>SAF1</td>
<td>TFT1</td>
<td>TFT1</td>
</tr>
<tr>
<td>bb33</td>
<td>0 \rightarrow 1</td>
<td>SAF0</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
<tr>
<td></td>
<td>1 \rightarrow 0</td>
<td>TFT1</td>
<td>SAF1</td>
<td>SAF1</td>
</tr>
</tbody>
</table>
4.4 Process-Voltage (PV) variations and fault models

We expected that PV variations can affect the fault behaviors of a synchronizer with different bridging defects. To check the hypothesis, we set up an experiment which uses a two-level approach instead of the direct worst-case process corner approach in [51], because we are more concerned with circuit correctness than circuit speed. We assumed that $V_{DD}$ is varied by 10% to reflect a variation of supply voltage due to switching activities. For $L$ and $W$, we set their standard deviation to 10% with regard to line edge roughness. The standard deviation for gate oxide thickness reserves 4.1 Å. At last, the standard deviation of transistor threshold voltage is assumed 10% for random dopant fluctuations and other effects. Table 4.7 gives the summary of all process and voltage variations in consideration. Then, according to Table 4.7, we performed Monte-Carlo simulations for the basic bridging structures in Section 4.1 (first level PV analysis). To observe the process variation effect in a 45nm or 65nm technology, we amplified the process variation effect for the considered 180nm technology to 10%, such that we can observe the fault behaviors successfully. From the simulation results, we chose the set of PV parameters with which a faulty synchronizer may show different fault behaviors from one with nominal PV parameters. Finally, by applying the set of PV parameters obtained, SPICE simulations for the synchronizer with 0Ω bridging defect were performed to explore more fault behaviors (second level PV analysis). The advantage of this two-level approach is to dramatically reduce the simulation efforts.

| Parameter | $\mu$ | $\sigma$ | | $\mu$ | $\sigma$
|-----------|-------|----------| |-------|--------|
| $W$       | $W$   | $W \times 10\%$ | | $L$   | 180 nm | 18 nm
| $T_{ox,n}$| 41 Å  | 4.1 Å    | | $T_{ox,p}$ | 41 Å | 4.1 Å |
| $V_{th,n}$| 0.369V| 0.035V   | | $V_{th,p}$ | -0.383V | 0.035V |
| $V_{DD}$  | 1.8V  | 0.18V    | |       |        |        |

For example, we performed Monte-Carlo simulations for the basic bridging structure in Fig. 4.2 by applying the PV variations in Table 4.7. For the case of nominal PV parameters, with $CLKB = 1$, the voltage on the shorted net is 1.387V (0.352V) when the applied input is logic 0 (logic 1). The variations of PV parameters result in different voltages on the shorted net. Table 4.8 gives the summary of the Monte-Carlo simulation performed. The mean voltage on the shorted net is almost the same as the voltage for the case of nominal PV parameters. But, the maximum and minimum voltages show significant differences. The maximum voltage is 1.796V (0.716V) and the minimum voltage is 0.932V (0.149V) when the applied input is logic 0 (logic 1). So, we expected that if these maximum and minimum voltages on the shorted net propagate to the down-stream circuits, the synchronizer may present varied fault behaviors.
Among all bridging defects in consideration, $bc_{13}^{DF1}$, $bc_{36}^{DF1}$, $bc_{13}^{DF2}$, and $bc_{36}^{DF2}$ can cause the same bridging structure as shown in Fig. 4.2 for the synchronizer under test. We have injected each defect into the synchronizer. Then, SPICE simulation for the faulty synchronizer is performed by applying the PV parameters which result in the maximum and minimum voltages on the shorted net. Table 4.9 presents varied fault behaviors of the faulty synchronizer for each bridging defect. The third column is the fault behaviors of the synchronizer with each bridging defect simulated using nominal PV parameters. The fourth (fifth) column is the fault behaviors of the synchronizer for the set of PV parameters which causes the maximum voltage on the shorted net, when the applied input is logic 0 (logic 1). The sixth column is the fault behaviors of the synchronizer for the set of PV parameters which causes the minimum voltage on the shorted net. In this case, the same PV parameter causes the minimum voltage on the shorted net no matter what input is applied. According to Table 4.9, fault behaviors of the synchronizer injected $bc_{36}^{DF1}$, $bc_{13}^{DF2}$, and $bc_{36}^{DF2}$ are altered by the applied PV parameters. However, all fault behaviors observed are still in the fault categories we found before. For other bridging structures, some PV variations result in different fault behaviors for the synchronizer; however, those fault behaviors are all in the fault categories of Table 4.1. Consequently, we did not find any new fault model introduced by PV variations.
Chapter 5

Analysis of Open defects of a two-D-flip-flop synchronizer

5.1 Open fault characteristics on locality

There are basic characteristics to comprehend fault behaviors of the synchronizer caused by open defects. First, we analyze open defects in a transmission gate and in an inverter, respectively. We emphasize that the all open defects considered in this section are completely opens, which means they have an infinite resistance.

5.1.1 Open defect in a transistor of a transmission gate

If the NMOS (PMOS) transistor of a transmission gate has an open defect, the signal still can propagate through the PMOS (NMOS) of the transmission gate. However, PMOS (NMOS) cannot propagate $0V (V_{DD})$ perfectly. When the drain of PMOS (NMOS) is driven by $0V (V_{DD})$ and PMOS (NMOS) is turned on, its source voltage can only reach to $|V_{T,p}| (V_{DD} - V_{T,n})$ instead of $0V (V_{DD})$. Moreover, it may need a longer time than a fault-free one. Such an open defect can cause a delay fault effect.

5.1.2 Open defect at the drain and source of an inverter

If a transistor of an inverter has an open defect at the source or drain, we intuitively expect that the path from the output of the inverter to $V_{DD}$ or the ground is disconnected and the output can keep its previous state. Of course, if there is an open defect in a NMOS ($OI_{5,5}$ or $OI_{5,7}$), the output finally goes to logic-0, when given enough time for the output to be discharged. For example, if a PMOS (NMOS) in the inverter has an open defect and a falling (rising) signal transition is applied, the output of the inverter retains logic-0 (logic-1, but will be logic-0 eventually due to leakage current) because the PMOS (NMOS) is disconnected from $V_{DD}$.
In addition, even though an open defect is located in the same transistor of an inverter, the synchronizer shows different fault behaviors depending on where the open defect is located. Fig. 5.1(a) shows the fault-free case. For simplicity, without consideration of wire capacitance and capacitive effect of the transmission gate next to inverter $i_2$, the load capacitance of $i_2$ includes gate-drain capacitance $C_{gd12} (= C_{gd1} + C_{gd2})$ caused by the Miller effect, and $C_{db1}$ and $C_{db2}$ which are the overlap capacitances of both $M_1$ and $M_2$ of $i_2$. The capacitances between drain and bulk in $M_1$ and $M_2$ are due to the reverse-based $pn$-junctions. We assume that a rising signal transition is applied when $CLKB = 1$, and $CLK$ becomes high before the input of $i_1$ is fully charged to $V_{DD}$, but high enough to be recognized as logic-1. For the case, the feedback loop formed by $i_1$ and $i_2$ when $CLK = 1$ makes the input of $i_1$ stabilized to logic-1.

If an open defect is located at the drain of the PMOS in inverter $i_2$ as shown in Fig. 5.1(b), $C_{db2}$ will be disconnected from the load capacitance of inverter $i_2$. Further $C_{gd12}$ will be replaced by $C_{gd1}$ due to the disappearance of the Miller effect. Thus, the effective load capacitor of $i_2$ becomes smaller than the fault-free inverter. If a rising signal transition is applied when $CLKB = 1$ and then $CLK$ becomes high before the input of $i_1$ is fully charged to $V_{DD}$ but high enough to be recognized logic-1, charge sharing occurs between the input of $i_1$ and the output of $i_2$. Because the effective load capacitance of $i_2$ is smaller, the input voltage of $i_1$ is dropped slightly, still high enough to keep logic-1 as its input. This makes the correct logic value propagate to the next latch of DFF1.

If an open defect resides at the source of the PMOS in inverter $i_2$ as shown in Fig. 5.1(c), $C_{db2}$ can be included to the load capacitance of $i_2$, but $C_{gd12} (= C_{gd1} + C_{gd2})$ resulting from the Miller effect will be smaller than the fault-free one because the inverter cannot work properly. However, the effective load capacitance induced is larger than that by the open defect at the drain of the PMOS (Fig. 5.1(b)). When $CLK = 1$, charge sharing occurs and leads to voltage drop at the input of $i_1$, larger than that of the case where the open defect is located at the drain, and the input voltage of $i_1$ becomes smaller than $V_{LTH}$ which is the logic threshold voltage of the inverter used. Finally, the feedback loop of $i_1$ and $i_2$ resets the input of $i_1$ to the previous state, logic-0. In the upcoming $CLKB = 1$, inverter $i_1$ can respond to its input change. Consequently, this causes a delay fault. Or, if the input is back to logic-0 before the upcoming $CLKB = 1$, inverter $i_1$ misses its previous input change.

5.1.3 Open defect at the input of inverter

Fig. 5.2 illustrates the case where the NMOS gate of an inverter has an open defect. When the input of the inverter is 0V, the PMOS of the inverter is turned on, and the output voltage of the inverter ($V_{out}$) is around (ground).
Figure 5.1: Open defects in the same transistor

(a) Fault-free

(b) Open defect at the drain

(c) Open defect at the source
$V_{DD}$. Based on the circuit structure of Fig. 5.2, the gate voltage of the NMOS can be derived by Equation 5.1 [41] where $C_{gd}$ ($C_{gs}$) is the capacitance between the gate and the drain (source) of the NMOS, and $C_{gb}$ is the capacitance between the gate and the bulk of the NMOS.

The values of capacitances $C_{gd}$, $C_{gs}$ and $C_{gb}$ depend on the state of the NMOS. In our HSPICE simulation, $V_{gs}$ becomes smaller than half $V_{out}$, but larger than $V_{T,n}$. Therefore, initially, there is a conducting path from $V_{DD}$ to the ground when the PMOS is turned on. When the input voltage of the inverter changes to $V_{DD}$, the PMOS is turned off, and the output voltage of the inverter is $V_{out}$ at that time. According to our HSPICE simulation, $V_{gs}$ for the NMOS determined by $V_{out}$ is high enough to turn on the NMOS for a short period. Thus, the $V_{out}$ value is reduced drastically due to discharging through the NMOS. When the $V_{gs}$ value of the NMOS is smaller than $V_{T,n}$, the NMOS turns off and the $V_{out}$ voltage drops slowly due to leakage current.

Fig. 5.3 illustrates the values of $V_{out}$ and $V_{gs}$ for the NMOS of the inverter with an open defect at the gate with a 180nm technology. The solid line is the $V_{gs}$ value of the NMOS, while the dotted line is the $V_{out}$ value of the inverter, respectively. Each waveform in the range of 0ns to 100ns is when the input voltage is logic-0. As shown, the output voltage $V_{out}$ is smaller than 1.8V because there is a conducting path from $V_{DD}$ to the ground through the faulty NMOS. When the input is changed to logic-1 at 100ns, the output voltage $V_{out}$ is discharged through the faulty NMOS drastically, and then reaches around 0.219V after 1µs. The case in which the PMOS gate of an inverter has an open defect can be analyzed in the same way. Such a fault can cause a delay fault as will be shown in Section 5.2. Tables 5.1 (5.2) presents the simulation result when an open defect is located at the gate of a PMOS (NMOS) with a falling (rising) transition. The first column gives the technology models considered, the second column is the threshold voltage of the PMOS (NMOS), and the third column is the initial gate voltage of the faulty PMOS (NMOS) when there is a conducting path from $V_{DD}$ to the ground. The fourth column gives the gate voltage of the faulty PMOS (NMOS) at 1µs after the fault-free NMOS (PMOS) is turned off. $V_{DD}$ for the inverter used is applied 1.8V for the 180nm technology model, and 1.0V for 45nm and 65nm technology models, respectively.

| PMOS | $|V_{T,p}|$ | $t = 0$ | $t = 1\mu s$ |
|------|-----------|--------|---------|
| 45nm | 0.203V    | 0.594V | 0.825V  |
| 65nm | 0.190V    | 0.573V | 0.877V  |
| 180nm| 0.522V    | 0.968V | 1.509V  |

If an inverter has an open defect at its input port as shown in Fig. 5.4, the voltage at the input is determined by topological characteristics, namely internal capacitances of the inverter, coupling capacitances to the surrounding nodes and the static and dynamic voltages on those coupled nodes, and the gate-oxide
Figure 5.2: NMOS with an open defect at the gate

Figure 5.3: Waveform of the gate with an open defect of NMOS
Table 5.2: Voltages at the gate with open defect of NMOS transistor

<table>
<thead>
<tr>
<th>NMOS</th>
<th>$V_{T,n}$</th>
<th>$t = 0$</th>
<th>$t = 1\mu s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>45nm</td>
<td>0.218V</td>
<td>0.459V</td>
<td>0.228V</td>
</tr>
<tr>
<td>65nm</td>
<td>0.212V</td>
<td>0.453V</td>
<td>0.089V</td>
</tr>
<tr>
<td>180nm</td>
<td>0.434V</td>
<td>0.653V</td>
<td>0.219V</td>
</tr>
</tbody>
</table>

trapped charge at the floating gate of the inverter [44]. However, the trapped charge is dependent on process technology and the coupling capacitances are dependent on the states of neighboring nodes. Equation 5.2 gives how to compute the inverter’s voltage at the input with an open defect. $C_{a1}$ and $C_{a0}$ are capacitances between the floating input and its neighboring nets which have logic-1 or logic-0. So, $C_{a1}$ and $C_{a0}$ are pattern-dependent. $C_{vdd}$ and $C_{gnd}$ are capacitances between the floating node and the power and ground rails, respectively. $C_{int}(1)$ and $C_{int}(0)$ are the internal capacitances inside the driven inverter. It is important to point out that $C_{int}(1)$ and $C_{int}(0)$ are not fixed, but directly depend on voltage $V_{in}$. Since the trapped charge at the floating node is unknown, $V_{trap}$ is unknown, too. Table 5.3 gives the floating input voltage and the resulting output voltage of the inverter designed by different technology models without consideration of coupling capacitances to the surrounding nodes. Such an open defect can cause two different fault effects, stuck-at fault and pulse fault, depending on the open input location as discussed in 5.2.1 and 5.2.4.

$$V_{gs} = \frac{C_1}{C_1 + C_2 + C_{gb}} \cdot V_{out}$$  \hspace{1cm} (5.1)

$$C_0 = C_{gnd} + C_{a0} + C_{int}(0)$$

$$C_1 = C_{vdd} + C_{a1} + C_{int}(1)$$  \hspace{1cm} (5.2)

$$V_{in} = \frac{C_1}{C_0 + C_1} \cdot V_{DD} + V_{trap}$$

Table 5.3: Voltages at the floating input of an inverter

<table>
<thead>
<tr>
<th>Technology model</th>
<th>Input Voltage</th>
<th>Output Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>45nm</td>
<td>0.557V</td>
<td>0.209V</td>
</tr>
<tr>
<td>65nm</td>
<td>0.509V</td>
<td>0.302V</td>
</tr>
<tr>
<td>180nm</td>
<td>0.883V</td>
<td>0.682V</td>
</tr>
</tbody>
</table>
5.2 Analysis of fault behaviors

In this section, we present examples to explain fault behaviors caused by injecting open defects for the synchronizer. Stuck-at fault (SAF), delay output fault (DOF), and pulse output fault (POF) are found for the synchronizer with open defects with infinite resistance. Delay output fault by resistive open defect (DOFR), and one-time pulse fault (OTP) are additionally observed for the synchronizer with resistive open defects.

5.2.1 Stuck-at fault (SAF)

Open defects can cause stuck-at-0 (SAF0) and stuck-at-1 (SAF1) faults for the synchronizer. If the input or output of a transmission gate \( t_1, t_3, t_5, \) or \( t_7 \) has an open defect, the fault behavior of the synchronizer is SAF0 or SAF1, because the node driven by the transmission gate becomes floating and the voltage of the floating node is set as explained in Section 5.1.3. In addition, if an open defect resides at the input or output of an inverter \( i_1, i_3, \) or \( i_5 \), the fault behavior of the synchronizer is also SAF0 or SAF1 due to the same reason as the transmission gates.

\( OT_{18} \) in Fig. 5.5 is given as an example. \( OT_{18} \) causes the output of transmission gate \( t_1 \) to be disconnected from node 2, and the input of inverter \( i_1 \) (node 2) to be floating. In our HSPICE simulation, the floating input voltage of inverter \( i_1 \) is 0.611V, and results in weak logic-1 as the inverter’s output as shown in Fig. 5.5(a).

When \( CLK = 1 \), the initial charge at the floating node is discharged through the NMOS transistor of inverter \( i_2 \) which is turned on by the weak logic-1, and the input voltage of inverter \( i_1 \) becomes logic-0 as shown in Fig. 5.5(b). Then, the synchronizer keeps logic-0 no matter what input value is (SAF0).

If an open defect is located at the input of inverters \( i_1, i_3, i_5, \) and \( i_7 \), the input voltages of those inverters...
are determined by internal capacitances of those inverters and surrounding factors as explained in Sec. 5.1.3. For example, Fig. 5.6 shows the synchronizer with $OI_{i_34}$. For the inverter designed by the 180nm TSMC technology model, the input voltage of an inverter ($i_3$) with a complete open defect is 0.882V and its output voltage is 0.682V. As a result, the synchronizer behaves like a SAF0.

![Figure 5.5: OT18 open defect](image)

![Figure 5.6: Synchronizer with $OI_{i_34}$](image)

### 5.2.2 Delay output fault (DOF)

For $OI_{i_{23}}$ and $OI_{i_{27}}$, the synchronizer responds its input signal application one clock later than expected, if the input signal is applied at $t_{\text{dist}} = -0.1\, \text{ns}$. The notation $t_{\text{dist}}$ stands for the time distance from the rising edge of the clock to an input signal application time. For example, if an open defect is located at the source of the PMOS transistor in inverter $i_2$ ($OI_{i_{23}}$) as shown in Fig. 5.7, the load capacitances of inverter $i_2$ include $C_{db_2}$ and $C_{gd_{12}}$ with $C_{gd_{12}}$ smaller than the fault-free one, as explained in Sec. 5.1.2. When a rising transition is applied at $t_{\text{dist}} = -0.1\, \text{ns}$, $t_1$ is turned off before the input of inverter $i_1$ is charged to 1.2V instead of fully charged to $V_{\text{DD}}$, but still high enough to be recognized as logic-1 as shown in Fig. 5.7(a). When $CLK = 1$ and $t_2$ is turned on, charge sharing occurs between the input of inverter $i_1$ and the output of inverter $i_2$. Because
of the charge sharing, the input voltage of inverter $i_1$ is slightly dropped, but inverter $i_1$ recognizes the input voltage as logic-0, unfortunately. Then, logic-1 of the output of inverter $i_1$ turns on the NMOS transistor of inverter $i_2$, and the input of inverter $i_1$ keeps logic-0 as shown in Fig. 5.7(b). In upcoming $CLKB = 1$, inverter $i_1$ can respond to its input change correctly due to the full charging process (i.e., the charge sharing effect does not fail the circuit function). As a result, the synchronizer with $O_{I_{23}}$ presents its output response one clock cycle later.

Fig. 5.7: Synchronizer with $O_{I_{23}}$

Fig. 5.8 shows the simulation result. For the fault-free synchronizer, when a rising transition is applied at $t_{dist} = -0.1\text{ns}$, the first D flip-flop responds it immediately. However, the synchronizer with $O_{I_{23}}$ responds it one clock later because of the reason explained above. Moreover, an open defect in the PMOS or NMOS transistor of a transmission gate can also cause a delay fault as explained in Section 5.1.1.

### 5.2.3 Delay output fault by resistive open defect (DOFR)

A synchronizer with an open defect at a certain resistance can generate its output several clocks later than expected, due to the extra propagation delay through the net with the resistive open defect. Fig. 5.9(a) shows the first latch of DFF1 where $i_1$ has an open defect with $27\text{k}\Omega$ resistance at its input. When a rising transition is applied and $CLKB = 1$, the voltage at node 2 becomes $V_{DD}$. However, node 3 cannot achieve the full $V_{DD}$ because of the open resistance. When $CLK = 1$, node 2 is driven back to logic 0 by the feedback loop shown in Fig. 5.9(a) for the first several clock cycles. However, node 3 is charged with higher and higher voltage as clock keeps switching as shown in Fig. 5.9(b). Finally node 3 can surpass $V_{LT\text{H}}$ which is the logic threshold voltage of the inverter, and inverter $i_1$ inverts its output successfully. HSPICE simulation shown in Fig. 5.9(b) shows that inverter $i_1$ inverts its output around four clock cycles later after the input
Figure 5.8: Simulation result of the synchronizer with $O_{I23}$
change. As a result, the faulty synchronizer generates its output four clock cycles later than expected.

(a) Synchronizer with $O_{I14}$ at 27kΩ

(b) Simulation result

Figure 5.9: $O_{I14}$ open defect

5.2.4 Pulse output fault (POF)

An open defect located at DFF2 can generate pulses for the faulty synchronizer depending on its input logic value. The synchronizer with $O_{I64}$ is given as an example in Fig. 5.10, which has an open defect at the input of inverter $i_6$. 
Fig. 5.10(a) shows the internal state of the synchronizer when its input is logic-0 and \( CLK = 1 \). The input of inverter \( i_6 \) is floating and the resulting output voltage of \( i_6 \) is 0.682V. When \( CLKB = 1 \), the 0.682V is fed to inverter \( i_5 \), which recognize as logic-0. So, the output of the synchronizer keeps logic-0. When the input is changed to logic-1 and the new output (i.e., logic-1) of DFF1 is applied to DFF2 in \( CLKB = 1 \) as shown in Fig. 5.10(b), the output of inverter \( i_5 \) is updated from logic-1 to logic-0. However, the input voltage of \( i_6 \) keeps 0.882V due to the open defect. In the upcoming \( CLK = 1 \), first the output of inverter \( i_5 \) (logic-0) is propagated to inverter \( i_7 \), whose output is changed to logic-1 as the output of the synchronizer as shown in Fig. 5.10(c). At the same time, the output voltage of inverter \( i_6 \) (0.682V) is propagated to inverter \( i_5 \), which recognize as logic-0. As a result, the output of inverter \( i_5 \) is set back to logic-1 and also propagated to inverter \( i_7 \). Thus, the output of the synchronizer is reset to logic-0 as shown in Fig. 5.10(d). In the next \( CLKB = 1 \), the input of \( i_5 \) is set to logic-1 again by the output of DFF1 as shown in Fig. 5.10(d). As a result, the faulty synchronizer generates pulses repeatedly at its output when its input is logic-1.

Figure 5.10: Synchronizer with \( OI_{64} \)
5.2.5 Undefined output fault (UOF)

An open defect can cause the output of the synchronizer to show a voltage in the range of $0.3 \cdot V_{DD}$ and $0.7 \cdot V_{DD}$. Open defects located at the input and output of inverter $i_7$ make the output voltage of the synchronizer in that range, as explained in Sec. 5.1.3. Fig. 5.11 presents the synchronizer with $OI_{74}$ which makes the input of inverter $i_7$ is floating. As a result, the input voltage of the floating inverter becomes $0.882V$ and the corresponding output voltage is $0.682V$ which is in the range of the undefined logic value. The logic response to the undefined voltage depends on the input logic threshold voltage of the downstream gates, which is the Byzantine effect.

![Figure 5.11: Synchronizer with $OI_{74}$](image)

5.2.6 One-time pulse fault (OTP)

An open defect at a certain resistance may cause a synchronizer to generate an unexpected one-time pulse. Fig. 5.12 shows the synchronizer with an open defect $OI_{84}$ at $158.7k\Omega$, and the synchronizer is implemented by a 180nm technology model. The faulty synchronizer is initialized by logic-0 at its input as shown in Fig. 5.12(a), and then its input is changed to logic-1, and latched to DFF2 in Fig. 5.12(b). When $CLK = 1$ in Fig. 5.12(b), the output of the first latch of DFF2 is propagated to the second latch. However, because of the resistor of the open defect, the input of inverter $i_8$ cannot reach the full zero voltage to invert its output. As a result, in the upcoming $CLKB = 1$ period, the input of inverter $i_7$ is driven to logic 1, and the output of the faulty synchronizer becomes logic-0, again (the voltage at the input of inverter $I_8$ remains at $0.603V$). In the next $CLK = 1$, the input voltage of inverter $i_8$ turns to $0.909V$, and the faulty synchronizer can keep logic-1 as its output. Thus, the faulty synchronizer shows an one-time pulse before its stable output is generated.

5.3 Summary of observed fault behaviors

We have performed an experiment on a synchronizer consisting of two D flip-flops by considering all possible open defects to check their potential fault behaviors. Table 5.4 shows the categories of all fault behaviors observed for the synchronizer. The first column in Table 5.4 gives the categories of all fault behaviors. The
Figure 5.12: Synchronizer with $O_{I_{84}}$
second column shows more detailed fault types belonging to each category, and the detailed fault behavior of each type is described in the third column. Each of the fault categories has been discussed in the previous section.

Table 5.4: Category of all open faults

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-At Fault</td>
<td>SAF0</td>
<td>stuck-at-0</td>
</tr>
<tr>
<td></td>
<td>SAF1</td>
<td>stuck-at-1</td>
</tr>
<tr>
<td>Delay Output Fault</td>
<td>DOF</td>
<td>Open defects cause delay faults</td>
</tr>
<tr>
<td>Delay Output Fault by resistive open defect</td>
<td>DOFR</td>
<td>A synchronizer with a resistive open defect generates its output more than one clock later than expected</td>
</tr>
<tr>
<td>Pulse Output Fault</td>
<td>POF</td>
<td>Open defects cause the faulty synchronizer to generate pulses.</td>
</tr>
<tr>
<td>One Time Pulse Fault</td>
<td>OTP</td>
<td>The faulty synchronizer shows an one-time pulse before its valid output is generated.</td>
</tr>
<tr>
<td>Undefined Output Fault</td>
<td>UOF</td>
<td>The output voltage of a faulty synchronizer is in the range of $0.3\cdot V_{DD}$ and $0.7\cdot V_{DD}$.</td>
</tr>
</tbody>
</table>

Table 5.5 presents the occurrence frequencies of fault behaviors obtained with all open defects implemented using an 180nm TSMC technology model. The first column in Table 5.5 shows the observed fault behaviors. The second column is the number of open defects for each fault behavior and the third column is its percentile. An open defect located in inverter $i_2$, $i_4$, $i_6$, or $i_8$ except its input port and transmission gate $t_2$, $t_4$, $t_6$ or $t_8$ (Fig. 3.1) is fault-free, because the data still can be memorized in a dynamic way at the input of $i_1$, $i_3$, $i_5$ or $i_7$. In addition, if an open defect is located in the PMOS transistor of $t_3$ or $t_5$, it works properly because the corresponding NMOS transistor still works well. Open defects in inverters $i_1$, $i_3$, and $i_5$ cause stuck-at faults (SAF). Open defects resided in both transistors of transmission gates $t_1$, $t_3$, and $t_7$ bring about delay output faults (DOF), while open defects at the input or output of those transmission gates result in SAFs. If an open defect is located at the input of inverter $i_7$, the output voltage of the faulty synchronizer is in the undefined logic region (UOF). Pulse output faults (POF) are generated for the synchronizer with $O_{I_{64}}$ and $O_{I_{84}}$ when its input is logic 1 and logic 0, respectively.

Table 5.5: Fault occurrence frequencies

<table>
<thead>
<tr>
<th>Fault type</th>
<th># of faults</th>
<th>Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-At Fault</td>
<td>39</td>
<td>30.47%</td>
</tr>
<tr>
<td>Delay Output Fault</td>
<td>22</td>
<td>17.19%</td>
</tr>
<tr>
<td>Undefined Output Fault</td>
<td>1</td>
<td>0.78%</td>
</tr>
<tr>
<td>Pulse Output Fault</td>
<td>2</td>
<td>1.56%</td>
</tr>
<tr>
<td>Fault-free</td>
<td>64</td>
<td>50.00%</td>
</tr>
</tbody>
</table>
To explore fault behaviors further, we used the *bisectional Pass/Fail* analysis function provided by HSPICE to search the resistance (called *sensitive resistance*) value of each open defect (from zero resistance) to which the synchronizer works properly. For the cases where open defects are located in DFF2, it is difficult to find the sensitive resistance value because even for a small resistance value, the faulty synchronizer generates its output with delay. So, if the synchronizer with an open defect at a certain resistance value gives its output 10% longer delay than the fault-free circuit, we assumed the resistance value as the sensitive resistance value for this open defect. Then, for each open defect \( f \), 15 different resistance values which are evenly distributed in the range of 0\( \Omega \) to 1G\( \Omega \) in logarithm scale are added one by one to the sensitive resistance of \( f \). Finally, these 15 resistive values are injected into open defect \( f \) of the synchronizer under test to further explore their new fault behaviors.

For the synchronizer with resistive open defects, we observed two different fault behaviors from the synchronizer with complete open defects: delay output fault by resistive open defect (DOFR) and one-time pulse fault (OTP). By increasing the resistance value of an open defect, the faulty synchronizer presents its output later than expected. The longest delay fault observed is 14 clock cycles later than expected for \( OI_{72} \) at around 4.033M\( \Omega \). The synchronizer with \( OI_{84} \) in the range of 158.709k\( \Omega \) and 217.823k\( \Omega \) shows an 1-0-1 one-time pulse fault (OTP) before its valid output for the rising signal transition.

Besides, we have performed the same experiment on the synchronizer implemented using 45nm and 65nm predictive technology models [50]. All fault behaviors observed are in the same fault categories obtained for the 180nm TSMC technology model. For 45nm and 65nm technology models, the number of delay faults are reduced in the given clock period of 2ns because the inverters and transmission gates implemented are faster. However, the major causes for the delay faults are still valid.
Chapter 6

Summary of fault behaviors and impact on system

In Chapters 4 and 5, we have investigated all possible fault behaviors of the two-D-flip-flop synchronizer with complete bridging and open defect as well as the case where the defects have finite resistance values. In comparison with each other, many of observed faults are the same: stuck-at fault (SAF), delay output fault (DOF), and undefined output fault (UOF). Some of them have different names, but the same fault behaviors. The delay output fault by resistive open defect (DOFR) shows the same fault behavior of TFT3 of functional timing fault of the synchronizer with bridging defect. The pulse output fault (POF) caused by the open defect belongs to POF1 of the synchronizer with bridging defect because the open defect causes the synchronizer to generate pulse output depending on the input value. However, one-time pulse fault for both defects has the same name, but different fault behaviors. Fig. 6.1 presents the difference of one-time pulse faults caused by bridging and open defect. When the input is logic-0, the synchronizer with the bridging defect keeps logic-1. When the input is changed to logic-1, the synchronizer goes to logic-0, and then becomes logic-1 finally. So, the bridge defect causes 1-0-1 one-time pulse fault (OTP0). However, for the synchronizer with the open defect, when the input is logic-0, the synchronizer outputs logic-0 also. When the input is changed to logic-1, the synchronizer outputs logic-1 at the expected time. However, the upcoming $CLKB = 1$, the output is reset to logic-0, and goes to logic-1 at the next rising edge of CLK.

With the consideration of all faults of bridging and open defects, we combine Table 4.1 and 5.4 into one table, Table 6.1. The DOFR and POF fault caused by open defects are merged to TFT3 and POF1 in the table, respectively. The one-time pulse fault of the open defect is added to the table as OTP2 of the one-time pulse fault category.

Table 6.1 presents fault categories of the two-D-flip-flop synchronizer with all possible bridge and open defects. Whether a fault in the synchronizer results in a failure in the system is determined by the fault type.
Table 6.1: Category of all bridging and open faults

<table>
<thead>
<tr>
<th>Category</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuck-At Fault</td>
<td>SAF0</td>
<td>Stuck-at-0</td>
</tr>
<tr>
<td></td>
<td>SAF1</td>
<td>Stuck-at-1</td>
</tr>
<tr>
<td>Functional Timing Fault</td>
<td>TFT1</td>
<td>The output of the synchronizer is observed earlier than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT2</td>
<td>The inverted output of the synchronizer is observed earlier than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT3</td>
<td>The output of the synchronizer is observed at least one clock cycle later than expected.</td>
</tr>
<tr>
<td></td>
<td>TFT4</td>
<td>The inverted output of the synchronizer is observed at least one clock cycle later than expected.</td>
</tr>
<tr>
<td></td>
<td>DOF</td>
<td>The output with delay of the synchronizer is observed.</td>
</tr>
<tr>
<td></td>
<td>IDOF</td>
<td>The inverted output with delay of the synchronizer is observed.</td>
</tr>
<tr>
<td>Pulse Output Fault</td>
<td>POF1</td>
<td>The synchronizer generates a pulse with regard to its input signal.</td>
</tr>
<tr>
<td></td>
<td>POF2</td>
<td>The synchronizer generates a pulse regardless of its input signal.</td>
</tr>
<tr>
<td>One-Time Pulse Fault</td>
<td>OTP0</td>
<td>The output of the synchronizer shows 1-0-1 one-time pulse.</td>
</tr>
<tr>
<td></td>
<td>OTP1</td>
<td>The output of the synchronizer shows 0-1-0 one-time pulse.</td>
</tr>
<tr>
<td></td>
<td>OTP2</td>
<td>The output of the synchronizer shows one-time pulse after valid signal transition</td>
</tr>
<tr>
<td>Internal Oscillation Fault</td>
<td>IOF</td>
<td>A signal oscillation occurs in the synchronizer by a loop, and the output of the synchronizer depends on the applied clock frequency.</td>
</tr>
<tr>
<td>Undefined Output Fault</td>
<td>UOF</td>
<td>The output voltage of the faulty synchronizer is in the range of $V_{DD} \cdot 0.3$ to $V_{DD} \cdot 0.7$.</td>
</tr>
</tbody>
</table>
and the circuit the faulty synchronizer drives. Stuck-at (SAF), pulse output (POF), one-time pulse (OTP), and internal oscillation (IOF) faults can result in a system error due to their dramatic change in the circuit behavior. Early-transition (TFT1 and TFT2) timing faults can bring about reliability issues by shortening the MTTF. Moreover, inverted-early-transition (TFT2) timing fault can result in a system failure by introducing the opposite logic value to a combinational circuit as shown in Fig. 6.2(a). Besides, early transition faults can result in a system failure as follows. In the case of a 2-way bundled data handshake-based synchronization mechanism, the earlier request signal through the faulty synchronizer can make data on the bus fetched before the valid data arrives. Late-transition (TFT3 and TFT4) timing faults lead to system performance degradation. Inverted-late-transition (TFT4) timing fault can result in a system failure with the same reason as TFT2. For delayed output (DO and IDO) faults, their fault effects depend on which circuit the faulty synchronizer drives. For example, if the faulty synchronizer drives the circuit as shown in Fig. 6.2(a) and one of the inverters drives a critical path, it can cause a propagation delay fault. If the faulty synchronizer drives the circuit in Fig. 6.2(b), the pulse width of the edge detector becomes shorter than expected because the faulty synchronizer outputs a signal transition with extra delay. For an undefined output fault (UOF), how the undefined value is interpreted depends on the circuit the faulty synchronizer drives. In Fig. 6.2(a), if the synchronizer outputs an undefined voltage, all three inverters might interpret it differently.

![Synchronizer driving combinational circuits](image1)

(a) Synchronizer driving combinational circuits

![Synchronizer driving an edge detector](image2)

(b) Synchronizer driving an edge detector

Figure 6.2: Example circuits driven by a synchronizer
Chapter 7

Asynchronous FIFO and test method development

In the current System-on-a-chip (SoC) era, the issue of safe and reliable data communication between synchronous and asynchronous cores has come to the fore and elaborate synchronization methods are required [6]. These clock domain crossing (CDC) signals are typically classified into two types (i) synchronous and (ii) asynchronous depending on whether there is a phase/frequency relationship between the receiving and sending domains [52]. While synchronous clock crossing signals are timed and robustly verified, asynchronous CDC signals cannot be verified accurately since the order of clock edges are flexible.

Fig. 7.1 shows an asynchronous FIFO synchronization design widely used in industry for reliable data communication between two asynchronous clock domains. According to [53], incorrectly implemented FIFO designs still work properly 90% of the time and almost-correct FIFO designs function properly 99%+ of the time. Design verification of FIFOs which work properly 99%+ of the time is usually the most difficult. In addition, it is difficult to test FIFOs that have timing faults caused by defects in synchronizers, because such timing faults do not lead to catastrophic function failure in test mode.

Fault models and test methods for different types of synchronous FIFOs have been presented in [54–56]. A BIST (built-in self testing) scheme for ring-address type FIFOs has been developed in [57], and a BIST method for embedded dual-port FIFOs has been proposed in [58]. In [59], an asynchronous test method for FIFOs has been proposed. For testing required timing specification between read-write operations, “delta-skew generator” is inserted to the write clock pin in series and read-write patterns are applied repeatedly while changing the position of the write pulse relative to the read pulse [59]. However, this test may not be applicable to the given asynchronous FIFO design shown in Fig. 7.1 because the FIFO works in asynchronous clock domains and it is difficult to define a specific timing specification between read-write operations.
Two-D-flip-flop synchronizers are not recommended to be replaced with scan cells because the multiplexer of each scannable D flip-flop can introduce extra delay and make the expected mean-time-to-failure (MTTF) of the synchronizer shortened. To test the synchronizer at rated speed, we need a special test capture scheme, like staggered single capture or staggered double capture scheme [60].

In this chapter, we propose a novel method for testing synchronizers in the asynchronous FIFO design shown in Fig. 7.1. For at-speed testing, we propose a test clock generator that consists of a launch clock generator, a path delay controller, and a capture clock generator for generating the required clock cycles at the desired time. Especially, the proposed capture clock generator has the flexibility of changing its capture clock frequency. Also, we propose a calibration method to guarantee the accuracy of the desired delay and the generated capture clock signal. In the next chapter an efficient test method is proposed to fully test the synchronizers in the FIFO based on fault models developed in the previous captures.

### 7.1 The Architecture of asynchronous FIFO design

There is an asynchronous FIFO design (ASYNC-FIFO) [53] widely used in industry for asynchronous CDC signal synchronization. Fig. 7.1 presents the block diagram of this ASYNC-FIFO. Here, the memory block is a dual-port RAM in which read and write operations can be performed simultaneously. The write (read) control block generates an address for the memory to write (read) data, and detects if the FIFO memory is full (empty).

To access the FIFO memory block, a binary-coded address \( \text{waddr} \) and \( \text{raddr} \) is used. Simultaneously,
the binary-coded address is converted into its corresponding gray-coded address ($wpt$ and $rpt$), and is then transferred to the other clock domain to determine the FIFO state, full or empty. The advantage of the gray-coded address transfer is that it eliminates the multi-bit signal change problem when it is latched into synchronizers. Fig. 7.2 presents the address generator of the write and read control block of the ASYNC-FIFO.

The write address ($waddr$ and $wpt$) always points to the next word to be written; therefore, by reset, both (binary-coded and gray-coded) addresses are set to zero, which is the next FIFO memory word location to be written. On a write operation, the memory location that is pointed by the address $waddr$ is written, and then the address is incremented for pointing to the next location to be written.

Similarly, the read address ($raddr$ and $rpt$) always points to the current FIFO memory word to be read. The $raddr$ address is pointing to invalid data because the FIFO memory is empty, and the empty flag is asserted to indicate this. As soon as the first data word is written to the FIFO memory, the address $waddr$ increments, and the empty flag is cleared. The $raddr$ address that is still addressing the first FIFO memory word drives the first valid word onto the FIFO memory data output port, to be read by the receiver logic.

To distinguish between full and empty, an extra bit is added to each address as the most significant bit (MSB). When a write operation performs past the last FIFO address, the newly added MSB of the write address ($waddr$) is used instead of back to the first address of the FIFO (i.e. “01111” to “10000”). The same is done for the read address ($raddr$). If the MSBs of two addresses are different, it means that the write address has wrapped one more time than the read address. Otherwise, both addresses has wrapped the same number of times. This fact makes the way of empty detection different from that of full detection for the FIFO design. For an ASYNC-FIFO, the empty and full detectors use the gray-coded addresses ($wpt$ and $rpt$) instead of their binary-coded addresses ($waddr$ and $raddr$). If $wpt$ and $rpt$ are the same in
gray-coded addresses, the FIFO is empty. If both first two bits of \textit{wptr} and \textit{rptr} are different and the rest of bits are the same, the FIFO is full by comparing 10111 and 01111. For example, if \textit{wptr} (\textit{waddr}) is “10111 (11010)” and \textit{rptr} (\textit{raddr}) is “01111 (01010)”, respectively, the FIFO is full. Fig. 7.3 shows a logic block that checks the state of ASYNC-FIFO and determines the full (empty) flag.

7.2 Test Scheme and Circuits

7.2.1 Test Structure

In the asynchronous FIFO design, there are two registers in each clock domain: \texttt{REG}\textsubscript{B} for binary-coded address and \texttt{REG}\textsubscript{G} for gray-coded address in each of both write and read control blocks shown in Fig. 7.1. These registers are scannable. However, synchronizers in the dotted boxes of Fig. 7.1 are recommended not to apply scan design because the added combinational circuit for scan design can reduce the Mean Time To Fault (MTTF) by changing the characteristic of these synchronizers. Fig. 7.4 shows the test structure to test the synchronizers in the read clock domain (TCK2) that has five address lines. The address saved in \texttt{REG}\textsubscript{G} of the write clock domain (TCK1) is sent to the TCK2 domain, and the synchronizers under test latch the address. Then, “Empty Detector” compares the sent address from the TCK1 domain with the one kept in the TCK2 domain. By checking the output of “Empty Detector”, we can evaluate the test result. To test synchronizers in the write domain, the read and write domains becomes TCK1 and TCK2, respectively. Also, “Empty Detector” in the above discussion is replaced by “Full Detector” to evaluate the test result.

As mentioned before, only registers (\texttt{REG}\textsubscript{B}, \texttt{REG}\textsubscript{G}, and D-flipflops for \textit{ef} and \textit{ff} flags) in the dotted boxes of Fig. 7.4 are scannable. A test scheme shown in Fig. 7.5 can be used to test the synchronizers in the TCK2 domain by the following steps:

1. Shift test data into scan chains in the TCK1 and TCK2 domains.

2. De-assert the scan enable (\textit{SE}) signal.
3. Apply at least two TCK2 cycles to initialize each synchronizer under test in the TCK2 domain by shifting test data (e.g., 11111) into REG_G in the TCK1 domain.

4. Apply a launch clock (LCLK) cycle to launch a gray-coded address (e.g., 10011) to the TCK2 domain by latching the new gray-coded address of Bin2Gray logic into REG_G in the TCK1 domain. This is used to create transitions for the synchronizers to sensitize their faults.

5. Apply three capture clock (CCLK) cycles to capture the output of “Empty Detector”. Two TCK2 clock cycles are required for the synchronizers under test to generate their outputs to “Empty Detector”. The third TCK2 clock cycle is used to latch the output of the empty detector into the empty flag flip-flop. The number of CCLK cycles can be different depending on the type of faults to detect.

6. Shift the test result out.

---

**Figure 7.4:** Test structure

**Figure 7.5:** Test scheme
7.2.2 Test clock generation for proposed test scheme

Fig. 7.6 presents the design of a test clock generator (TCK_GEN) for the test scheme given in Fig. 7.5. TCK_GEN consists of a delayed SE signal generator (DSE_GEN), a launch clock generator (LCLK_GEN), a capture clock generator (CCLK_GEN), and a programmable delay (PD) element to implement \( d_{\text{adjustable}} \). The FCK1 and FCK2 are the functional clock signals of the TCK1 and TCK2 domains, respectively. As will be explained later, the FCK1 signal is used to generate a launch clock signal, and the FCK2 signal as a reference clock signal when a pseudo-functional clock signal generated by CCLK_GEN is calibrated.

7.2.3 Programmable delay (PD) element

First of all, we present a PD element [61] that is used to emulate the channel delay between two clock domains and to generate a pseudo-functional clock (which is not a real functional clock) signal during the test mode. The PD element consists of a chain of delay buffers (coarse- and fine-grained buffers) and a scannable shift register to scan in a value (C) to control the delay. A delay buffer can be modeled as shown in Fig. 7.7(a) for convenience. The delay buffer of Fig. 7.7(a) can be either a coarse-grained delay buffer or a fine-grained delay buffer, and Fig. 7.7(b) is its block diagram. Fig. 7.8 shows a PD element consisting of delay buffers and a scannable shift register. The delay of the PD element is determined by turning on or off different delay buffers. So, the control data length is \( n+m \) where \( C[n+m-1]...C[m] \) control the coarse-grained delay buffers and \( C[m-1]...C[0] \) control the fine-grained delay buffers.
7.2.4 Channel delay compensator (CH_CMP)

In the ASYNC-FIFO, gray-coded addresses are clock domain crossing (CDC) signals as shown in Fig 7.9(a). Their propagation delay $d_{\text{channel}}$ must be provided as a design specification. When a gray-coded address is launched from register REG_G in the TCK1 domain, the sent address will be latched by the synchronizers in the TCK2 domain with delay $d_{\text{channel}}$. So, for testing the synchronizers in the TCK2 domain, when a gray-coded address is launched from the TCK1 domain, the first capture clock cycle must arrive at the synchronizers in the TCK2 domain at least $d_{\text{channel}}$ later (shown in Fig. 7.5).

We use the set of the programmable delay (PD) element discussed above to emulate the $d_{\text{channel}}$ propagation delay by adjusting the $d_{\text{adjustable}}$ delay of the channel delay compensator (CH_CMP) as shown in Fig. 7.9(b). Additionally, $d_{\text{adjustable}}$ can have different values depending on the test method to detect a specific fault. In general, without consideration of gate delays in the circuit in detail (assuming that $d_{\text{channel}}$ is much larger than the gate delay in CCLK_GEN), $d_{\text{adjustable}}$ must satisfy the following equation
to guarantee that synchronizers in the TCK2 domain latch the issued gray-coded address from the TCK1 domain. Such a correlation is depicted again in Fig. 7.9(b).

\[
d_{adjustable} \geq d_{channel} + d_{tck1} - d_{tck2} \tag{7.1}
\]
7.2.5 Delayed scan enable signal generator (DSE_GEN)

As mentioned before, after the SE signal is de-asserted, at least two TCK cycles are required for all synchronizers in the TCK2 domain to be initialized by the data saved in REG_G of the TCK1 domain. To implement this function, we use a delayed scan enable (DSE) signal. When the SE signal is de-asserted, the DSE signal (used in TCK2 selection shown in Fig. 7.6) will be de-asserted a certain time later (channel propagation delay plus at least two TCK clock cycles) to guarantee that all synchronizers in the TCK2 domain can be initialized by the address sent from the TCK1 domain. Fig. 7.10 presents the implementation of the delayed SE (DSE) signal generator. The delay element to implement $d_{\text{channel}}$ (i.e., the delay between REG_G of TCK1 and the synchronizers of TCK2) shown in Fig. 7.10(a) can be either the PD element introduced above or any delay element that can give at least $d_{\text{channel}}$ of delay. When the SE signal is logic 1, all D-flipflops of the DSE signal generator are reset so that signal DSE follows the SE of logic 1. When the SE signal is de-asserted, the output (logic 1) of the first D-flipflop in the DSE_GEN block is propagated through the delay element to the cascaded D-flipflops, and then DSE is de-asserted. When SE signal is asserted again, the DSE signal generator is asserted immediately as shown in Fig. 7.10(b).

In Fig. 7.6, the select signal of the multiplexer for TCK1 domain is the SE signal. When the SE signal is asserted, the TCK clock is fed to the TCK1 domain for test data shifting. When the SE signal is de-asserted, a LCLK cycle generated by LCLK_GEN is provided to the TCK1 domain. However, the LCLK cycle is generated after the DSE signal is de-asserted because the LCLK_GEN is triggered by DSE signal (Fig. 7.6 and 7.11). As a result, no clock cycle is fed to the TCK1 domain until the DSE signal is de-asserted. For the
TCK2 domain, the select signal of the multiplexer is the DSE signal. So, after the SE signal is de-asserted, the test clock (TCK) is still applied to the TCK2 domain while the DSE signal keeps logic 1 to initialize all synchronizers in the TCK2 domain. When the DSE signal is de-asserted, LCLK_GEN generating a LCLK signal and a copy of the LCLK (LCLK\textsubscript{d} shown in Fig. 7.6) signal penetrating through the CH_CMP with \(d_{\text{ad}}\) just able to activate CCLK\_GEN to generate the CCLK signal for the TCK2 domain.

**Launch clock generator (LCLK\_GEN)**

Fig. 7.11 presents the implementation of LCLK\_GEN which passes a cycle of functional clock FCK\textsubscript{1} when the DSE signal is de-asserted (Fig. 7.11(b)). The generated clock signal is used to launch a test data from register REG\_G in the TCK1 clock domain to the TCK2 clock domain. A copy of the launch clock (LCLK\textsubscript{d} in Fig. 7.6) signal is propagated through CH_CMP with \(d_{\text{ad}}\) just able to activate CCLK\_GEN.

**7.2.6 Capture clock generator (CCLK\_GEN)**

Fig. 7.12 presents the implementation of CCLK\_GEN triggered by the LCLK\textsubscript{d} signal that is a copy of LCLK signal from LCLK\_GEN (Fig. 7.6) penetrating through the CH_CMP with delay \(d_{\text{adj}}\). By adjusting \(d_{H}\) and \(d_{T}\), the capture clock (CCLK) signal can be generated. The CSEL signal shown in Fig. 7.12 selects how many CCLK cycles will be generated depending on the type of faults to detect. Note that DFF5 and DFF6...
are dual-edge triggered D-flipflops.

Fig. 7.12(b) presents the case where CCLK_GEN generates two CCLK cycles by CSEL = 1. The rising edge of LCLK\textsubscript{d} triggers the rising edge of the first CCLK cycle through double-edge triggered DFF5. The falling edge of CCLK is set off by the rising edge of LCLK\textsubscript{d} through the d\textsubscript{H} PD element. Also, the rising (falling) edge of the first CCLK cycle through the d\textsubscript{T} PD element triggers double-edge triggered DFF6 that generates the rising (falling) edge of the next CCLK cycle. As a result, CCLK\_GEN generates two CCLK cycles with time period d\textsubscript{T}. More CCLK cycles can be generated by giving an appropriate value to CSEL. Here, we assume that d\textsubscript{H} and d\textsubscript{T} are both much greater than gate and flip-flop delays.

Most of clock triggering circuits for at-speed testing [60] are fed with clock signals generated by the phase-locked loop circuit on a system. These schemes have a drawback such that the delay between the launch clock and the capture clock is not controllable because they are asynchronous. Therefore, they do
not fit well for asynchronous at-speed testing.

The proposed CCLK_GEN circuit makes it possible to generate a pseudo-functional clock signal with different frequencies by adjusting $d_H$ and $d_T$, which increases the reusability of the circuit. In addition, with a PD element for emulating the channel delay, it is easy to control the delay between the launch clock and the capture clock. This is the major reason why a functional clock (e.g., FCK1 in LCLK_GEN) cannot be used in CCLK_GEN.

### 7.3 A calibration method for programmable delay element

![Figure 7.13: Calibration flow chart](image)

The behavior of a faulty two-D-flipflop synchronizer depends on the input application time and the frequency of the applied clock signal. This fact calls for at-speed testing for synchronizers in the ASYNC-FIFO. For at-speed testing, it is necessary to set $d_{adjustable}$, $d_H$, and $d_T$ in Fig. 7.9 and Fig. 7.12(a) respectively to specific values. As explained before, those delays are implemented by the PD elements introduced in Section 7.2. The resolution of an implemented delay is determined by the smallest delay step size which
is the delay of a fine-grained buffer. However, all buffers are exposed to process variation, and the delay realized by a PD element might not be the same as the desired delay.

7.3.1 Calibration process

Wang et al. proposed a high-accuracy calibration process in [62] that uses a TDF test method. We adopted it to calibrate the PD elements for $d_{\text{adjustable}}$ in CH_CMP as well as $d_H$ and $d_T$ in CCLK_GEN. Fig. 7.13 shows the modified calibration process. First, a control vector is shifted into the scanable shift register as shown in Fig. 7.8, and a transition delay fault (TDF) test is performed for the PD element under calibration with a specified delay. The test result is compared to the previous one. If the test result passes and the test output is the same as the previous one, the calibration circuit computes a new control vector which generates a longer delay. If the test result passes (fails) and the test output is the same as the previous one, the calibration circuit computes a new control vector which generates a longer (shorter) delay. Repeat the test until the test result is different from the previous one. If the test result is different from the previous one, the calibration circuit compares the current control vector with the previous one. If only the last bit (C[0]) is different, the calibration process is finished. Otherwise, the calibration circuit generates a new control vector between the current control vector and the previous one, and the entire process is repeated as shown in Fig. 7.13.

7.3.2 Calibration of channel delay compensator with $d_{\text{adjustable}}$

The channel delay compensator (CH_CMP) with $d_{\text{adjustable}}$ is utilized to emulate the $d_{\text{channel}}$ delay between two clock domains. To detect some types of faults (especially, earlier-than-expected fault [63]), $d_{\text{adjustable}}$ of CH_CMP needs to be calibrated to a specific delay value as will be presented later. To calibrate the CH_CMP, one transmission gate (CTR1) is added to the LCLK_GEN logic block, and another transmission gate (CTR2) and one D-flipflop (CDFF1) are added to the CCLK_GEN logic block as shown in Fig. 7.14(a). As introduced in Section 7.3.1, first a control vector (C) is shifted into the control vector register of the CH_CMP to set the desired delay value, and the “CAL” signal is asserted. Then, two TCK cycles are applied as shown in Fig. 7.14(b). The first TCK cycle gives a launch cycle and it is propagated through the CH_CMP. The rising edge of the propagated cycle triggers DFF4 (which was initialized to logic 0) in the CCLK_GEN block. The second TCK cycle is used to latch the output of DFF4 into CDFF1. If CDFF1 latches logic 1 (logic 0), the delay of the CH_CMP set by the control vector might be shorter (longer) than expected. Then, we shift in another control vector which makes the delay caused by the PD element inside of the CH_CMP longer (shorter), and repeat the same process shown in Fig. 7.13 until the CH_CMP has the desired delay value.
7.3.3 Calibration for capture clock generator

The CCLK_GEN block generates a specific number of pseudo-functional clock signals for the clock domain under test. For example, if synchronizers in the read (write) clock domain are under test, the CCLK_GEN block provides pseudo-functional rclk (wclk) clock signals (Fig. 7.1) to that clock domain. Two PD elements in the CCLK_GEN block make it possible to generate these pseudo-functional clock signals: one for the high-level clock period ($d_H$) and the other for the entire clock period ($d_T$). So, our calibration method is divided into two phases. In the first phase, the PD element for $d_H$ is calibrated, and then the PD element for $d_T$ is calibrated.

Reference clock generator

As shown in Fig. 7.11, the LCLK_GEN block generates a launch clock signal directly using functional clock signal FCK1. However, instead of using functional clock signal FCK2, the CCLK_GEN block generates capture clock signals indirectly by adjusting the PD elements for $d_H$ and $d_T$ triggered by the launch clock signal for testing. A reference clock signal is required to calibrate the $d_H$ and $d_T$ values such that the clock signals generated by the CCLK_GEN block and by the functional clock signal of the clock domain can be as similar as possible. Fig. 7.15(a) shows the implementation of a reference clock generator. Here, signal sel_func_clk decides which clock signal is used as a reference, i.e., the rclk (wclk) signal if synchronizers in the read (write) clock domain are under test. Further, the CSE signal is similar to the SE signal except
that it is useful to scan the control vectors. When signal CSE has logic 1, a control vector is shifted into the PD element of $d_H$ or $d_T$ under calibration. When signal CSE is de-asserted to logic 0 which indicates that shifting in a control vector is finished, the reference clock generator generates one or two cycles of the functional clock signal (FCK2) for the clock domain under test. As will be presented later, one cycle of the reference clock signal is needed to calibrate the PD element for $d_H$ ($CAL_T = 0$), but two cycles of the reference clock signal are required to calibrate the PD element for $d_T$ ($CAL_T = 1$) instead. Fig. 7.15(b) shows the waveforms of the reference clock generator.

**First phase - Calibration of the PD element for $d_H$**

Fig. 7.16(a) shows a part of CCLK_GEN which generates logic 1 of the desired capture clock signal. A multiplexer to select between $LCLK_d$ and $FCK2$, a transmission gate (CTR3) and a D-flipflop (CDFF2) are added for calibration. Fig. 7.16(b) shows the calibration scheme. Signal $CCK$ is the clock for shifting in control vectors to the PD element for $d_H$ and can be TCK. $CSE$ signals is the shift enable signal for
calibration. \textit{FCK2} is the reference clock signal fed by the reference clock generator as shown in Fig. 7.15(a). As explained in Section 7.3.1, first of all, when \textit{CSE} has logic 1, a control vector is shifted into the control flip-flops of the PD element to set the time period for the high logic value (i.e., \(d_H\) in Fig. 7.12(b)) of a capture clock with the desired delay. Then, apply a clock cycle using the reference clock signal (FCK2) as shown in Fig. 7.16. The rising transition of FCK2 triggers DFF4 to generate a rising transition. At the same time, the generated rising transition propagates through the PD element with delay \(d_H\) to generate the falling transition of the \textit{CCLK} signal. These rising and falling transitions bring about the rising and falling edge of the \textit{CCLK} signal by activating the double-edge triggered D-flipflop DFF5. The time period of logic 1 of the \textit{CCLK} signal is determined by the delay \(d_H\) of the PD element and the delay from all related flip-flops and gates (the dotted line). If the programmed delay for \(d_H\) is too short (long), \(B_{H\text{-short}}\) (\(B_{H\text{-long}}\)) at circuit node B will be generated as shown in Fig. 7.16(b), and CDFF2 keeps logic 0 (logic 1) as given in \(COUT_{1H\text{-short}}\)
Figure 7.17: Calibration of the PD element of $d_T$

$COUT_{1H-long}$, when it is triggered by the falling edge of FCK2 signal. Since delay $d_H$ is relatively small, the gate delays through the dotted line must be considered as part of $d_H$. The calibration circuit naturally implements this idea. When the entire calibration process terminates shown in Fig. 7.13, the delay of $d_H$ can be close to that of FCK2 depending on the resolution of the control vector.

**Second phase - Calibration of the PD element for $d_T$**

Fig. 7.17(a) shows a part of CCLK\_GEN which generates consecutive capture clock cycles. By adjusting $d_T$, the desired time period of each complete capture clock cycle can be achieved. Fig. 7.17(b) shows the
Figure 7.18: Integrated calibration circuit for CCLK_GEN

calibration scheme. As explained in Section 7.3.1, when CSE has logic 1, a control vector is shifted into the PD element to set the desired time period $d_T$ of a complete capture clock cycle. The time period of a full capture clock cycle is the delay of the closed loop (the dotted line) as shown in Fig. 7.17(a). When CSE is de-asserted, two consecutive clock cycles of the reference clock FCK2 are applied. The first reference clock cycle of FCK2 is used to trigger DFF4 to generate a capture clock cycle. The second reference clock cycle is used to capture (by CDFF2) the second capture clock cycle generated to measure the time difference between both rising edges of these two capture clock cycles generated. The scenario of $d_T$ calibration can be discussed as that for $d_H$ by observing the value of COUNT1. Again, the delays of gates and flip-flop involved in the entire loop must be (and have been) considered as part of $d_T$.

The integrated calibration circuit for CCLK_GEN

As observed in the previous sections, the required circuits to calibrate the PD elements for $d_H$ and $d_T$ are identical except the edge direction of the clock signal transition for CDFF2. To calibrate the PD element for $d_H$ ($d_T$), the falling (rising) edge of the reference clock signal (FCK2) is used. Fig. 7.18 presents the integrated circuit for calibrating both PD elements. When 'CAL_T' has logic 0 (1), the output of OR1 is latched by CDFF2 at the falling (rising) edge of FCK2.
Chapter 8

Test pattern generation for proposed test method

In this section, we briefly introduce all possible faults of a two-D-flipflop synchronizer resulting from bridging and open defects. Then, for synchronizers in the ASYNC-FIFO, we present test methods for detecting these faults.

8.1 Initialization test

The initialization test is designed to test whether the synchronizers under test can be initialized to the desired logic values. This test is so simple and yet so powerful that it can detect many faults before elaborate tests for specific faults are performed. Prior to the initialization test, all programmable delay elements ($d_{\text{adjustable}}$, and $d_H$, $d_T$ for CCLK_GEN) need to be set to the specific delay values. The $d_{\text{adjustable}}$ of CH_CMP must be long enough to guarantee that the input patterns arrive at the synchronizers under test earlier than the first capture clock cycle. Delay values for $d_H$ and $d_T$ of CCLK_GEN do not have to be stringent because the initialization test is not frequency-related. For simplicity, this research provides only two initialization test vectors, all zeros (0-initialization test) and all ones (1-initialization test).

Fig. 8.1 shows the 0-initialization test by which all synchronizers in the TCK2 domain are first set to all zeros using the test scheme shown in Fig. 7.5. Here, the test vector that will be used to set all synchronizers under test to logic 0 is shifted into the scan chain (REG_B and REG_G) in the TCK1 domain, and another test vector that makes Bin2Gray of the TCK2 domain generate all zeros is shifted into the scan chain (REG_B) in the TCK2 domain as shown in Fig. 8.1(a). After the initialization clocks by TCK2 are applied, all synchronizers should be set to logic 0 as shown in Fig. 8.1(b). Then, using the test scheme in
Fig. 7.5, a launch clock is applied, and the REG\_G in TCK1 still keeps all zeros. The output of “Empty Detector” in the TCK2 domain is latched into its following D-flipflop for the ef flag (ef flip-flop) by one capture clock cycle. For example, if the third synchronizer under test fails to be initialized to logic 0, the ef flip-flop latches logic 0 instead of logic 1 because the faulty value causes “Empty Detector” to generate logic 0 as shown in Fig. 8.1(b) (x/y means good machine/bad machine values). Note that the launch signal in TCK1 for initialization test can be removed out of the test scheme in Fig. 7.5. It is kept there to avoid changing the test scheme.
The 0 (1)-initialization test can detect any synchronizer with a stuck-at-1 (0) fault. In addition, inverted-earlier-than-expected (TFT2), inverted-later-than-expected (TFT4), inverted delay output (IDOF), and one-time pulse (OPT0 and OPT1) can also be detected by the initialization test because these faults make each faulty synchronizer initialized to the opposite logic value expected. Each undefined output fault (UOF) which causes any faulty synchronizer to output unknown logic value or behave as a stuck-at fault can be detected by the initialization test as well.

Table 8.1 presents the necessary conditions for delay elements and test patterns for the initialization test for testing synchronizers in an ASYNC-FIFO. We assume that the scan chain is built in the order of \textit{ef (ff)}-flip-flop, \texttt{REG.G}, and \texttt{REG.B} for read (write) clock domain. Note that \texttt{REG.G} in TCK2 clock domain is not shown in Fig. 8.1, since TCK2 is used for read and \texttt{REG.G} is not used in this example.

The TCK2 (TCK1) test pattern given in Table 8.1 is shifted into the \textit{ef (ff)}-flip-flop of the scan chain. In the case of "X0000000000000" ("XXXXX0000000"), the first five zeros go to the \texttt{REG.B} in TCK1 (TCK2), the next five zeros ("XXXXX") to \texttt{REG.G} in TCK1 (TCK2), and ‘X’ to the \textit{ff (ef)}-flip-flop in the write (read) clock domain. For the initialization test shown in Table 8.1, the test pattern for 0-initialization (1-initialization) test is given in the first (second) row of the initialization test. For 0-initialization test, test pattern "X00000000000" is used by TCK1 which is working as a write domain, and "XXXXX0000000" is applied to TCK2 which is working as a read domain. However, when TCK1 (TCK2) is changed to read (write) domain, the test patterns (X00000000000) and (XXXXX0000001) are applied, respectively. The same discussion can be given for 1-initialization test. The fifth column gives the number of capture clock cycles for each test. For the initialization test, only one capture clock cycle is required for the \textit{ef (ff)}-flip-flop to latch the output of “Empty (Full) Detector”. The last column presents the list of faults detectable by the given test.

### 8.2 Pulse output fault (POF) detection

According to the previous chapters, a resistive bridging or open defect in a faulty synchronizer can cause a pulse output (POF) fault by continuously changing the state of internal signals due to signal conflict or an internal signal feedback loop created by the defect. There are two different types of POF faults, POF1 and POF2, depending on whether the POF fault is related to an applied input value as shown in Table 6.1. If there is a test pattern to detect a POF1 fault which depends on an applied input value, it also can detect a POF2 fault which is independent of its applied input value. The frequency of a POF is the fraction (e.g., 1/2, 1/3) of the applied clock frequency to the synchronizers. Before performing the POF testing, it is a
Table 8.1: Test of synchronizers in the read (write) domain

<table>
<thead>
<tr>
<th>Step</th>
<th>Type</th>
<th>Test patterns</th>
<th># of CCLKs</th>
<th>Detected faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration #1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{adj ,stable} \leq d_{channel, min} + t_{setup} + d_{tck1} - d_{tck2}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_T$ is long enough. $d_T = 2 \cdot d_H$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization test</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-initialization</td>
<td></td>
<td>X000000000000 (X00000000000)</td>
<td>1</td>
<td>SA1, TFT2, TFT4, IDOF, OPT0, UOF</td>
</tr>
<tr>
<td>1-initialization</td>
<td></td>
<td>X111110101 (X111110101)</td>
<td>1</td>
<td>SA0, TFT2, TFT4, IDOF, OPT1, UOF</td>
</tr>
<tr>
<td>Pulse output fault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 1 transition</td>
<td></td>
<td>X0000010101 (X0000010101)</td>
<td>8</td>
<td>POF1, POF2, IOF, OPT2</td>
</tr>
<tr>
<td>1 to 0 transition</td>
<td></td>
<td>X1111100000 (X1111100000)</td>
<td>8</td>
<td>POF1, POF2, IOF, OPT2</td>
</tr>
<tr>
<td>Calibration #2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_T = d_T / 2$ $d_T$ is the time period of one clock cycle of the functional clock of TCK2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay test</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 1 transition</td>
<td></td>
<td>X0000010101 (X0000010101)</td>
<td>3</td>
<td>DOF, TFT3</td>
</tr>
<tr>
<td>1 to 0 transition</td>
<td></td>
<td>X1111100000 (X1111100000)</td>
<td>3</td>
<td>DOF, TFT3</td>
</tr>
<tr>
<td>$TFT_{CLK=0}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 1 transition</td>
<td></td>
<td>X000010101 (X000010101)</td>
<td>2</td>
<td>TFT1 fault activated</td>
</tr>
<tr>
<td>1 to 0 transition</td>
<td></td>
<td>X1111100000 (X1111100000)</td>
<td></td>
<td>during CLK = 0</td>
</tr>
<tr>
<td>Calibration #3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_{adj,stable} \leq d_{channel, min} + d_{tck1} - h_{hold}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d_H \geq d_{channel, min} - d_{tck1} - h_{hold}, d_T = 2 \cdot d_H$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TFT_{CLK=1}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 1 transition</td>
<td></td>
<td>X0000010101 (X0000010101)</td>
<td>3</td>
<td>TFT1 fault activated</td>
</tr>
<tr>
<td>1 to 0 transition</td>
<td></td>
<td>X1111100000 (X1111100000)</td>
<td></td>
<td>during CLK = 1</td>
</tr>
</tbody>
</table>

prerequisite that delays for $d_H$ and $d_T$ for CCLK Gen are calibrated to generate capture clock signals as shown in Calibration #1 in Table 8.1.

Fig. 8.2 shows an example of POF fault detection. Assume the third synchronizer in the TCK2 domain (read clock domain) has a POF1 that is triggered by its input with logic 1. The POF1 fault causes the synchronizer to generate pulses whose frequency is half of $f_{TCK2}$. The pulses also cause “Empty Detector” to generate pulses instead of steady logic 1. By monitoring the output of “Empty Detector” (“Full Detector”), we can confirm whether there exists a POF fault among the synchronizers under test.

Fig. 8.3 shows a test scheme to detect POF faults. In the initialization phase, the synchronizers under test are initialized to all zeros. Then, a launch clock (LCLK) cycle is generated by TCK1 to send all ones to TCK2 domain as shown in Fig. 8.2, and the delayed LCK will trigger the CCLK Gen circuit. The “CSEL” signal of CCLK Gen is used to control the number of capture clock cycles generated (Fig. 7.12). According to [64], the slowest frequency of pulses generated by a POF fault is $f_{CLK} / 3$ in which $f_{CLK}$ is the frequency of the applied clock signal. To detect such a fault, at least three capture clock cycles are required because the faulty synchronizer will generate the falling edge of the first pulse in the third capture clock, and the falling
edge can be detected by a pulse detector explained later. The proposed CCLK.Gen can generate up to eight capture clock cycles as shown in Fig. 7.12 and 8.3, and it is easy to modify the circuit to generate even more capture clock cycles. Fig. 8.3 shows the case where a POF fault causes the faulty synchronizer to generate pulses with frequency $f_{CLK}/2$ instead of steady logic 1. As a result, “Empty Detector” also generates pulses as its output response.

Fig. 8.4 presents a pulse detector that monitors the $e_{val}$ ($f_{val}$) signal to determine whether there is an unexpected pulse. The pulse detector is reset while the DSE signal has logic 1 and is activated by the rising edge of the first CCLK cycle. If an unexpected falling edge at the output of “Empty (Full) Detector” is generated, the edge triggers PDFF1 in Fig. 8.4 to output logic 1 instead of logic 0. Note that this pulse fault detection method depends on the output of “Empty (Full) Detector”. So, if multiple synchronizers have the pulse faults, the detectability of these faults using this method depends on the response of “Empty (Full)
detector”. This method also detects an internal oscillation fault (IOF) if it causes the faulty synchronizer to generate pulses. For OTP2 fault, according to the forth waveform of Fig. 6.1, the faulty synchronizer shows an unexpected one-time pulse after the second capture clock cycle, which also can be detected by the proposed pulse output fault detection method. Table 8.1 (row pulse output fault under Calibration #1) presents the test patterns to generate 0 to 1 and 1 to 0 transitions for the synchronizers under test.

![Figure 8.4: Pulse Fault detector](image)

### 8.3 Delay output fault (DOF) detection

A resistive bridging or open defect in a synchronizer can cause a delay fault. Such a delay fault raise a path delay fault for the critical path of following circuit. To detect the delay fault, at-speed testing is required. Prior to performing the delay fault test, $d_T$ and $d_H$ need to be calibrated by comparison with the functional clock of TCK2 domain. In addition, $d_{\text{adjustable}}$ should be adjusted to guarantee that a launched test pattern from TCK1 domain arrives at the input of the synchronizer under test in TCK2 domain before the first capture clock cycle. Note that this test method is for detecting delay fault caused by a signal conflict or a signal delay inside of the synchronizer under test, which is independent of the input application time.

To test the delay faults for the set of synchronizers simultaneously, we use test patterns which generate “00000” to “11111” (“11111” to “00000”) signal transitions. The REG_B register in TCK2 domain have a test pattern which generates the gray-code of “11111” (“00000”). So, “Empty (Full) Detector” of good machine generates logic 1. If one of the synchronizers under test has a delay fault and the delay is long enough, “Empty (Full) Detector” could not generate logic 1 when the third capture clock cycle is arrived. As a result, $ef (ff)$-flipflop will latch logic 0 instead of logic 1 as shown in Fig. 8.5. Note that first two capture clock cycles are used for the synchronizers under test to latch the launched test pattern. Table 8.1
Figure 8.5: Delay Fault detection

(row delay test under Calibration #2) presents the necessary conditions for delay elements and test patterns. The test patterns for DOF detection is the same as those for POF detection except the number of capture clock cycles.

This method also can detect later-than-expected (TFT3), additionally. TFT3 fault causes the faulty synchronizer to respond its input change at least one clock cycle later than expected. As a result, “Empty (Full) Detector” cannot generate logic 1 in a given time, and the following ef (ff)-flipflop keeps logic 0 at the third capture clock cycle.

8.4 Earlier-than-expected (TFT1) fault detection

Occurrence of a TFT1 fault depends on the arrival time of an input signal for the synchronizer under test. For example, if the input of a faulty synchronizer with defect bc12_{DF F1} (bc35_{DF F1}) has a rising or falling signal transition when the clock is high (low), the faulty synchronizer responds one clock cycle earlier than expected; otherwise it works like a fault-free synchronizer [63]. There are two different conditions to activate the earlier-than-expected timing fault depending on when an input signal is applied: CCLK = low or CCLK = high.

Fig. 8.6(a) shows the timing diagram to activate a TFT1 fault for CCLK = low. To activate the fault, the input of the synchronizer must receive the launched test pattern (with time = \(d_{\text{tck1}} + d_{\text{channel, max}}\)) before the rising edge of the first CCLK clock (i.e., TCK2) with enough setup time of the synchronizer (with time = \(d_{\text{adjustable}} + d_{\text{tck2}} - t_{\text{setup}}\)). As explained in Section 7.2, it is possible to control the time when the synchronizer
under test latches its input by changing the arrival time of the capture clock signal. If the PD element for $d_{adj,\text{justable}}$ (Fig. 7.8) is calibrated to satisfy Eq. 8.1, it is guaranteed that the applied test pattern arrives at the input of the synchronizers under test before the rising edge of the first capture clock cycle.

\[ d_{adj,\text{justable}} \geq d_{\text{channel,max}} + t_{\text{setup}} + d_{\text{tck1}} - d_{\text{tck2}}, \tag{8.1} \]

where $d_{\text{channel,max}}$ is the maximum channel delay, $d_{\text{tck1}}$ the delay time of the launch clock signal from LCLK_GEN to REG_G in TCK1 domain, $d_{\text{tck2}}$ the delay time of the capture clock signal from CCLK_GEN to the synchronizer under test in TCK2 domain, and $t_{\text{setup}}$ the setup time of the synchronizer under test, respectively. To detect the TFT1 fault for CCLK = low, the time period of the CCLK cycle (i.e., the values of $d_T$ and $d_H$) does not need to be stringent. Moreover, only two capture cycles are required to detect this fault.

For example, to detect an earlier-than-expected fault for a synchronizer in the read clock domain (TCK2), “X0000010101” and “1XXXXX00000” are the test patterns for scan chains in TCK1 and TCK2, respectively. Fig. 8.7(a) shows the state of the synchronizers after the initialization process is finished. “Empty Detector” outputs ‘1’. Then, the write clock (TCK1) domain launches “11111” by one LCLK cycle. For the good machine, the application of two CCLK clock cycles makes the synchronizers in TCK2 domain latch...
the address from TCK1 and feed to “Empty Detector” which outputs ‘0’. However, the output of “Empty Detector” has not been latched into its following D-flipflop (ef), which keeps ‘1’ as shown in Fig. 8.7(b). On the contrary, if one of the synchronizers has an earlier-than-expected fault, the faulty synchronizer outputs ‘1’ earlier than expected. Then, “Empty Detector” will generate ‘0’ earlier than expected and be latched to the D-flipflop by the end of the second CCLK cycle as shown in Fig. 8.7(c).

To activate the TFT1 fault activated when CCLK = high, a launched test pattern from TCK1 must arrive at the input of the synchronizer in TCK2 domain while CCLK is high as shown in Fig.8.6(b). If \(d_{\text{adjustable}}\) satisfies Eq. 8.2 shown below, the launched test pattern will arrive at the synchronizers in TCK2 domain after the rising edge of the first CCLK clock cycle. Note that \(t_{\text{hold}}\) is the hold time of the synchronizer under test. Moreover, if Eq. 8.3 holds, all signal transitions from TCK1 domain will arrive at the synchronizers while the first CCLK cycle is high. So, if \(d_H\) and \(d_{\text{adjustable}}\) follows Eq. 8.2 and Eq. 8.3 simultaneously, it is guaranteed that the launched test pattern arrives at the synchronizers in the TCK2 domain while the first CCLK cycle is high. The maximum (minimum) value of \(d_{\text{adjustable}}\) (\(d_H\)) is presented in Eq. 8.4 (8.5).

\[
\begin{align*}
\text{Eq. 8.2:} \quad d_{\text{adjustable}} &\leq d_{\text{channel, min}} + d_{\text{tck1}} - d_{\text{tck2}} - t_{\text{hold}} \\
\text{Eq. 8.3:} \quad d_H &\geq d_{\text{channel, max}} - d_{\text{adjustable}} + d_{\text{tck1}} - d_{\text{tck2}} \\
\text{Eq. 8.4:} \quad d_{\text{adjustable}} &\equiv d_{\text{channel, min}} + d_{\text{tck1}} - d_{\text{tck2}} - t_{\text{hold}} \\
\text{Eq. 8.5:} \quad d_H &\equiv d_{\text{channel, max}} - d_{\text{channel, min}} + t_{\text{hold}}
\end{align*}
\]

Further, because \(d_{\text{adjustable}}\) controls the time when the first capture clock cycle arrives at the synchronizers under test, by shortening \(d_{\text{adjustable}}\) and enlarging \(d_H\) simultaneously, the range of CCLK = high can be expanded to guarantee that the test pattern sent from TCK1 domain arrives at the synchronizers in the range of CCLK = high. Compared to the TFT1 fault activated when CCLK = low, this case requires three CCLK cycles. The launched address from TCK1 domain must arrive when the first CCLK cycle is high, and the synchronizers under test requires two more CCLK cycles to latch the address to feed to “Empty (Full) Detector”. In Table 8.1, the required conditions for \(d_{\text{adjustable}}\), \(d_H\), and \(d_T\) and test patterns are presented in the row of Calibration #3.
Figure 8.7: Earlier-than-expected timing fault detection
8.5 Undefined output fault (UOF) detection

Resistive bridging defects and resistive open defects of a synchronizer can cause an undefined logic value fault by which the output voltage of a faulty synchronizer is in the range of $0.3 \cdot V_{DD}$ to $0.7 \cdot V_{DD}$. The logic response of a circuit to the undefined logic value input depends on the input logic threshold voltage of downstream gates by the Byzantine effect. Even though the initialization test is able to detect some of UOF fault, an extra circuit is required for UOF fault detection.

If the synchronizer has an undefined output fault and its voltage is steady, the output of “Empty Detector” or “Full Detector” can be stuck at logic 0, logic 1, or undefined logic value. Assume that the third synchronizer in TCK2 domain of Fig. 8.7 has an undefined logic value fault. When “11111” is sent from TCK1 domain, the synchronizers output “11U11” due to the undefined logic value fault. The interpretation in the voltage of the undefined logic value depends on the input logic threshold voltage of “Empty Detector” with three possibilities:

1. If the detector evaluates the undefined logic value as logic 1, it behaves like fault-free. In this case, we need to apply another test pattern of “11011” to inspect how the detector works for the undefined logic value fault caused by logic 0 input value.

2. If the detector evaluates it as logic 0, the detector outputs logic 0 which is different from the fault-free result.

3. If the output of the detector is still in the range of the undefined logic value, it still shows as an undefined logic value fault.

We can test all synchronizers in a domain simultaneously by setting the synchronizers to all zeros (ones). If the detector outputs logic 1, it is fault-free. However, the output voltage of one of the synchronizers is in the range of undefined logic value, and “Empty (Full) detector” interprets it as the opposite, then the detector generates logic 0 instead of logic 1. This is the same as the initialization test as shown in Section 8.1. However, if the output of the empty detector has undefined logic value, we need a undefined voltage detector. Paisley [65] introduced a voltage window detector which monitors the voltage of a signal to decide whether it belongs to a certain range of a voltage as shown in Fig. 8.9. We can set $V_{REF1}$ and $V_{REF2}$ to $0.3 \cdot V_{DD}$ and $0.7 \cdot V_{DD}$, respectively by controlling the $R_1$, $R_2$ and $R_3$ values. Then, if the output of the empty detector is within the range of $V_{REF1}$ and $V_{REF2}$, the output of the voltage window detector becomes logic 1. Otherwise, it is logic 0. Further, such an undefined logic value fault might cause the downstream gates to
have longer delay to respond to their input applications, and then bring about delay fault effect. Therefore, this kind of undefined faults can be detected by the delay fault detection method introduced in Section 8.3.

Figure 8.9: Voltage window detector circuit

8.6 Summary of test methods and test procedure

As explained at the beginning of this section, bridging and open defects in the two-D-flipflop synchronizer cause several different faults: stuck-at (SAF1, SAF2), functional timing (TFT1, TFT2, TFT3, TFT4, DOF, IDOF), pulse output (POF1, POF2), internal oscillation (IOF), one-time pulse (OTP0, OTP1, OPT2), and undefined output (UOF) faults as listed in Table 6.1. Table 8.1 presents a sequence of steps for testing
synchronizers in the read (write) domain of ASYNC-FIFO. The first column shows the test and calibration procedures, and the second column lists the required test vectors. The third column presents the number of capture clock cycles for the corresponding tests. The last column lists detected faults by the test and test vector.

Before the initialization test is performed, $d_{\text{adjustable}}$, $d_H$, and $d_T$ need to be set to a desired value through a calibration procedure. Because the initialization test is not related to the frequency of the capture clock signal, $d_H$ can be long enough and $d_T$ is $2 \cdot d_H$. $d_{\text{adjustable}}$ also can be long enough to guarantee that the applied input transition arrives at the synchronizers in TCK2 when CCLK is low. The initialization test can detect SAF0, SAF1, TFT2 (inverted earlier than expected), TFT4 (inverted later than expected), IDOF (inverted delay output) OPT0 (1-0-1 one-time pulse), OPT1 (0-1-0 one-time pulse), and undefined output (UOF) faults. Then, the pulse output fault detection test is performed by applying eight capture clock cycles. As mentioned before, the slowest frequency of the pulse output fault in [66] is $f_{\text{CLK}}/3$, but it depends on the resistance value of an open defect. We apply eight capture clock cycles to detect slower pulse output fault. Through this test, the pulse output faults (POF1, POF2) and the internal oscillation fault can be detected. Note that the pulse output fault detection test does not need a calibration procedure. Before the delay fault detection test, $d_H$ and $d_T$ must be calibrated to be the same as the frequency of the functional clock of TCK2. For this test, three capture clock cycles are applied. At the last capture cycle, the output of the empty (full) detector is latched into its following D-flipflop. This test can detect delay output (DOF) and later than expected (TFT3) faults. To detect a TFT1 fault activated in CCLK = 0, the calibration procedure is not needed because the requirement is that the applied input transition must arrive at the synchronizers in TCK2 when CLK is low. Only difference is that the number of capture clock cycles is two. Because of the TFT1 fault, the empty (full) detector evaluate the input change earlier than expected and its output is latched into the following D-flipflop at the first or second capture cycle. On the other side, to detect TCK1 fault activated in CCLK = 1, $d_{\text{adjustable}}$, $d_H$, are $d_T$ are calibrated to guarantee that the applied input must arrive at the synchronizers in TCK2 when CCLK is high. The number of capture clock cycles should be three because the first capture cycle is used to activate the fault and the rest of capture cycles are used to detect the fault. Beside, note that a voltage window detector and a pulse fault detector are required additionally to detect an undefined output fault and pulse output fault, respectively. Test patterns for testing the synchronizers in the write clock domain are presented in parentheses under the test patterns for the synchronizers in the read clock domain. Moreover, test vectors for testing synchronizers in the read and write domains are not symmetrical because of the difference of the method to detect empty and full states.
Chapter 9

Conclusion and future research

In the first part of this thesis, we have presented the fault analysis and modeling of a two-D-flip-flop synchronizer. Initially, a bridging (open) defect with zero (infinite) resistance is injected into each pair of nodes of the synchronizer to maximize the fault effect. Most of faults are identified through the simulation with those complete bridging and open defects. Critical resistances are then found to search extra possible faults resulting from resistive bridging (open) defects. Also, we have performed experiments with different manufacturing technologies and process variations. Simulation results show that bridging and open defects can cause the synchronizer to generate stuck-at fault, functional timing fault, pulse output fault, one-time pulse fault, internal oscillation fault, and undefined output fault. In addition, fault behavior of the synchronizer depends on the location and resistance value of each defect, the input signal pattern (rising and falling), the input signal application time relative to the applied clock signal, and the applied clock frequency. However, applying scan design to synchronizers is not recommended because scan design may degrade their expected mean-time-to failure (MTTF).

In the second part, the research is extended to develop a test method for applications such as a FIFO to detect faults observed in the synchronizer. An asynchronous FIFO design that has been widely used for data communication between asynchronous IP cores is chosen as the candidate application. We assume that scan design is not applicable to synchronizers. A test method including typical stuck-at test and pseudo-asynchronous at-speed delay test with the support of additional test circuits is developed. To realize the idea of pseudo-asynchronous at-speed testing, a new capture clock generator is proposed. A well-designed calibration method is also proposed for calibrating delay elements of the capture clock generator, the delayed scan enable generator, and the channel delay compensator to desired delay values.
9.1 Contribution of this work

9.1.1 Fault modeling of two-D-flip-flop synchronizer

There are many research works for fault modeling and analysis of sequential circuits: latches and D-flip-flops. In this research, two-D-flip-flop synchronizers that are widely used for reliable data communication in asynchronous clock domains are considered. For a synchronizer, both complete and resistive bridging and open defects are thoroughly investigated to find its all possible fault behaviors. In addition, process and voltage variations in different technologies are considered to search for more faults. After massive experiments, several unusual fault effects are observed: stuck-at fault, timing related fault, pulse output fault, internal oscillation fault, one-time pulse fault, and undefined output fault. The results of this thesis help designers and test engineers analyze failures of a system that implements data communication using synchronizers: a handshake protocol and an asynchronous FIFO design (for example).

9.1.2 Test for two-D-flip-flop synchronizers

As mentioned before, scan design cannot be applied to two-D-flip-flop synchronizers due to the impact of their expected mean-time-to-failure (MTTF). In addition, the fault behaviors of defective synchronizers are different depending on several conditions. As a result, the current structural test method is not appropriate for the synchronizers. In this research, a hybrid test method is proposed to test the synchronizers of an asynchronous FIFO design. This method makes use of the scan design already applied to the FIFO control blocks, and performs a functional test for the synchronizers. Moreover, only six set of test patterns are required to test all synchronizers in both of read and write clock domains of the asynchronous FIFO design.

9.1.3 Pseudo-asynchronous at-speed testing technique

For current at-speed testing, clock signals are provided by phase-locked loop (PLL) logic, which means these clocks are synchronous. According to fault behaviors observed in this thesis, the need of an asynchronous at-speed testing is clearly mentioned because the fault behavior can be different depending on the input signal application times. As a solution, a pseudo-asynchronous at-speed testing method is proposed in this thesis. The proposed testing method is implemented by a newly designed capture clock generator and several delay emulators. The capture clock generator enables to generate pseudo-functional clock signals by configuring its frequency using delay elements. In addition, the path delay emulator is used to control the input signal application time. By using this scheme, the capture clock frequency and the input signal application time...
are flexible. In addition, a well-designed calibration method for these capture clock generator and delay emulators guarantees to set delay elements to desired values in spite of process variations.

9.2 Future work

A few possible research directions that can be pursued in the future have been outlined in this section.

1. Expansion of fault behaviors for general synchronizers

In this research, the two-D-flip-flop synchronizer implemented with transmission gates and inverters is used to categorize its possible fault behaviors. Even though this type of synchronizers is widely used in industry, there are many different types of synchronizers available. In addition, for fast synchronization, several sequential circuits (e.g., jamb latch, Q-flip [67]) specialized for synchronization are proposed. For the future research, we need to analyze them to further categorize fault behaviors. Because these synchronizers are fundamental in reliable data communication for asynchronous clock domains, it is required to perform precise failure analysis in the system level to categorize their fault behaviors. Because of the continuity of resistance value, we cannot guarantee that the fault categories we observed in this research cover all fault behaviors of the two-D-flip-flop synchronizer (though we tried to obtain all possible fault behaviors by using HSPICE, bisectional analysis considering different resistance values, manufacturing technologies, and process variation). For the future research, we may need to design a better experiment for this issue.

2. Extension of pseudo-asynchronous at-speed testing to general asynchronous at-speed testing

In the current state-of-art design era, globally asynchronous and locally synchronous (GALS) design is inevitable, which means current synchronous at-speed testing is not appropriate for testing GALS design. In this work, a pseudo-asynchronous at-speed testing is proposed. However, the proposed at-speed testing method is implemented depending on deep understanding of a fault behaviors of a two-D-flip-flop synchronizer and function of an asynchronous FIFO design. Thus, the proposed asynchronous at-speed testing method is specialized only for testing two-D-flip-flop synchronizers in the asynchronous FIFO design. As a future research topic, it may need to evolve to a general asynchronous at-speed testing method for GALS applications.
Bibliography


