Digital Testing with Multi-Valued Logic Signals

by

Baohu Li

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 9, 2015

Keywords: Multi-Valued Logic, test application, multi-site test reduced-pin-count test

Approved by

Vishwani Agrawal, James J. Danaher Professor of Electrical and Computer Engineering
Adit Singh, James B. Davis Professor of Electrical and Computer Engineering
Bogdan Wilamowski, Professor of Electrical and Computer Engineering
Victor Nelson, Professor of Electrical and Computer Engineering

Abstract

The integrated circuit scaling has been following the Moore's Law since 1965 [59, 60, 61]. Within these decades, researchers made great effort to shrink the transistor feature size and maximize the integration level. Together with the growing level of integration, the burden of testing continues to becoming heavier: the number of bits in each test pattern is larger because of increasing number of flip-flops and primary inputs; and larger number of test patterns is required to guarantee fault coverage in a large design. Both factors result in enormous test data volume. Although it is not a problem to generate a big test set by using automatic test pattern generation (ATPG) tools, it puts a higher demand on automatic test equipment (ATE) and prolongs test time, which increase testing cost rapidly. Nowadays, the testing related cost has taken about half of the total manufacturing cost. The industry urges to find a solution.

Built-in self-test (BIST) technology is investigated. It's frequently adopted in some particular cases like memory test to reduce or remove ATE cost. However, for more general testing purpose, BIST technology cannot guarantee fault coverage because of drawbacks like the pseudo-randomness of BIST patterns, which is hard to detect random-resistant faults and these faults aren't rare in large designs. An alternative thinking is to reduce test data volume but maintain the test information for same fault coverage. Fortunately most bits in test are don't-cares, which makes it possible to achieve high test compression level while retaining the fault coverage. Till now, test compression technologies have been successfully utilized in industry, compressing test size by up to 100x. Currently, the test compression interface exists in most large scale designs.

Improved test planning may utilize ATE and design-for-testability (DFT) resources wisely to reduce testing cost. Some useful metjods are concurrent test, multi-site test, hierarchical test and reduced-pin-count test (RPCT). These techniques can shorten test time or increase test throughput. Many solutions work together to limit the test cost within a reasonable level. For example, more and more test compression tools start to support low pin count test mode, which is equivalent to RPCT. On the other hand, RPCT technology enables more chips to be tested in parallel using multi-site test.

In current RPCT implementations, the test speed has to be the test channel data rate divided by the width of deserializer interface. The width of deserializer interface can be large in large system on chip (SoC) devices that support test compression. It turns out that the test speed needs to be very slow to match the deserializer interface with the limited test channel data rate. Traditionally, ATE test channel sends and receives binary data, in which case the data rate equals to the clock frequency. In this thesis, a revolutionary approach is introduced to improve ATE test channel efficiency so that test speed can be greatly increased especially for RPCT designs. My approach is to replace the media of test channel into multi-valued logic (MVL) signal. Here I choose digital to analog converter (DAC) as the MVL generator in the ATE test channel, and analog to digital converter (ADC) as the MVL decoder in the device under test (DUT). For MVL signal, multiple bits of information are contained within one symbol. So, an MVL link can send multiple bits of information in one clock cycle. As a result, it boosts the data rate of each physical test channel by several times. This property makes it suitable for the RPCT technology. The use of MVL interface can help resolve the problem of reduced test speed.

However, the use of MVL channel comes with additional sources of error over what we have in a binary system. For example, the insufficient accuracy of MVL generator and decoder, and larger vulnerability to noise. To tackle these, a new test

flow including ADC nonlinearity calibration and error detection and correction are developed. A prototype MVL test setup based on NI ELVIS II and Altera DE2 [5] FPGA is implemented to show the practicality of the MVL data transmission and to prove the concept of MVL test application. Further, I succeeded in generating the MVL signal from the available resources in the ATE system and used it to test a DUT with integrated MVL and RPCT interfaces. The experimental result shows that the use of MVL test channel greatly improves the test speed.

Acknowledgments

First and foremost, I would like to thank Dr. Vishwani Agrawal, who has advised me for years in Auburn. Without his priceless and useful guidance throughout my study and research, I would not be able to finish this dissertation. His attitude towards research and analytic thinking in problem solving sets me a good example for my Ph.D study and future career. Also, I would like to thank Dr. Adit Singh for being my committee member, together with his VLSI Testing course and suggestive advice on my work. I thank Dr. Bogdan Wilamowski for serving as my committee member and for adding to my understanding through his course. My thanks to Dr. Victor Nelson for his Computer-Aided Design course and instructions on tester, which were critical to my experimental implementation. I also thank him for going through my dissertation and polishing the language. I would like to thank to Dr. Xiao Qin for being my university reader.

I would also like to thank to the staffs in the department: Mary Lloyd, Shelia Collis and Linda Allgood for their assistance with different issues on campus; Joe Haggerty's assistance in experimental components supply.

I owe gratitude to my friends Bei Zhang, Hairuo Xu, Xinyu Zhu, Yang Xu, Zhou Hai and Ziqing Xu for years of company and help.

At last, I would like to thank my family for their support all the time. This work is dedicated to them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AMS	Analog and Mixed-Signal
ADC	Analog-to-Digital Converter
ADK	ASIC Design Kit
ATS	Applied Test Signature
ATPG	Automatic Test Pattern Generator
ATE	Automatic Test Equipment
BIST	Built-In Self-Test
CDF	Cumulative Distribution Function
CAR	Correct Application Rate
DNL	Differential NonLinearity
DAC	Digital Analog Converter
DUT	Device Under Test
DFT	Design For Testability
DM	Digital Module
DPS	Device Power Supply
EDT	Embedded Deterministic Test
FPGA	Field-Programmable Gate Array

FC Fault Coverage

GSI Giga-Scale Integration

IC Integrated Circuit

ITRS International Technology Roadmap for Semiconductors

INL Integral NonLinearity

LFSR Linear Feedback Shift Registers

LPCT Low-Pin Count Test

LSB Least Significant Bit

LSI Large-Scale Integration

MISR Multiple-Input Signature Registers

MVL Multi-Valued Logic

MSI Medium-Scale Integration

NI ELVIS II+ National Instruments Electronic Virtual Instrumentation Suite II Plus

OPTL Open Architecture Test Programming Language

PAM Pulse Amplitude Modulation

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

PSK Phase-Shift Keying

PI Primary Input

QAM Quadrature Amplitude Modulation

RPCT	Reduced-Pin-Count Test
RAM	Random-Access Memory
SSI	Small-Scale Integration
SSO	Simultaneous Switching Outputs
SOC	System on Chip
SiP	System in Package
SerDes	Serializer/Deserializer
SER	Symbol Error Rate
SNR	Signal Noise Ratio
TSV	Through Silicon Via
TAM	Test Access Mechanism
TRS	Test Response Signature
TPG	Test Pattern Generator
ULSI	Ultra-Large-Scale Integration
VLSI	Very-Large-Scale Integration

Chapter 1

Introduction

1.1 Very Large Scale Integration (VLSI) Technology

Since 1947, when the first transistor, a point-contact pn junction, was invented in Bell Labs, human history has stepped into a new era: the information age. In 1960, the first integrated circuit (IC), consisting of a single transistor and few passive components, was born in Fairchild, and became the prototype of all ICs we have today. As technology is developing, more and more transistors can be integrated into one chip. In 1965, Gordon Moore brought up the famous Moore's Law [59, 60, 61] to predict the IC industry development speed: transistor count in a single chip doubles every 18 months. Now we use different terms to describe the integration scale of ICs: small scale integration (SSI), medium scale integration (MSI), large scale integration (LSI), very large scale integration (VLSI), ultra large scale integration (ULSI) and giga scale integration (GSI). The component count for each integration level is shown in Table 1.1, which is available in [34].

However, we have become used to calling it VLSI technology no matter how large the integration scale goes. Technically, according to the function and structure, integrated circuits can be classified into digital, analog and analog and mixed-signal (AMS) ICs. As the name indicates, a digital IC is based on a digital circuit, which implements logic functions like numerical calculation, signal processing and data storage, etc. An analog IC it deals with continuous analog signals, doing jobs like amplifying, filtering and voltage conversion, etc. AMS ICs contain both digital blocks and analog blocks; examples include data converters and transceivers.

Table 1.1: Increasing complexity with generations of IC.

Scale	Components Count	Year
SSI (Small Scale Integration)	< 100	1963
MSI (Medium Scale Integration)	100-1000	1970
LSI (Large Scale Integration)	1000-10000	1975
VLSI (Very Large Scale Integration)	$10000 - 10^9$	1980
ULSI (Ultra Large Scale Integration)	$> 10^6$	1990
GSI (Giga Scale Integration)	$> 10^{10}$	2010

1.2 VLSI Testing

VLSI testing has been a part of VLSI technology from the very beginning. From the design stage to mass production, errors and mistakes are always happening. We are getting failed parts even with the most mature process technology. Today, the VLSI testing technology has been well developed, and various simulation and verification processes have been invented. In the typical IC product life cycle, the circuit needs to be first built on a wafer, then cut off and a chip is placed into a package. The end product will be set on a printed circuit board (PCB) and finally integrated into certain system. Errors may occur at every stage due to human mistakes (incorrect design and mishandling, etc.) and non-human factors such as random events during the production process or process-voltage-temperature (PVT) variations. Because it is more expensive to correct an error when it is detected at later stages, industry needs its IC products to be tested after every process stage. For example, there are wafer test and package test during fabrication, on-board test and system test after being passed to customers.

Usually, VLSI testing is performed until good chips are sent to customers, including wafer test, die test and package test. According to [26], in the aspect of testing objectives, VLSI test contains *parametric test* and *functional test*. Parametric Test

includes DC parametric tests such as leakage current test, etc., and AC parametric tests like delay test and setup/hold time test, etc. As for functional test, stuck-at fault test and behavioral functional test are representatives.

VLSI testing is not just giving some inputs to the circuits and checking the responses. There are lots of hidden work and intelligence involved. For example, the area of developing fault models tries to extract mathematical models (stuck-at faults, bridging faults and transition faults, etc.) from real hardware defects. With fault models, studies on test generation and fault simulation are required to obtain adequate test vectors. To support VLSI testing in hardware, different design for testability (DFT) technologies are developed to gain controllability and observability (through scan-based test), or to generate test stimulus (BIST), or to manipulate test data and responses (test data decompressor and test response compactor), etc. Because of the tremendous amount of money we spend on VLSI testing, it has become one of the most important branches in the IC industry. The developing VLSI technology pushes the requirements on testing to its limit all the time. Next, I'll introduce more details about recently emerging or still under exploration VLSI testing challenges.

1.3 Current Challenges in VLSI Testing

In Test and Test Equipment section of International Technology Roadmap for Semiconductors (ITRS) 2011 [16], the drivers for test technology are labeled as "device trends" like increasing device integration/interface bandwidth and 3D-stacked devices, etc., "increasing test process complexity" as caused by adaptive test and concurrent test, and "continued economic scaling of test" due to test parallelism, test data volume and interface/socket hardware costs management. These aspects well represent what problems we have in VLSI testing.

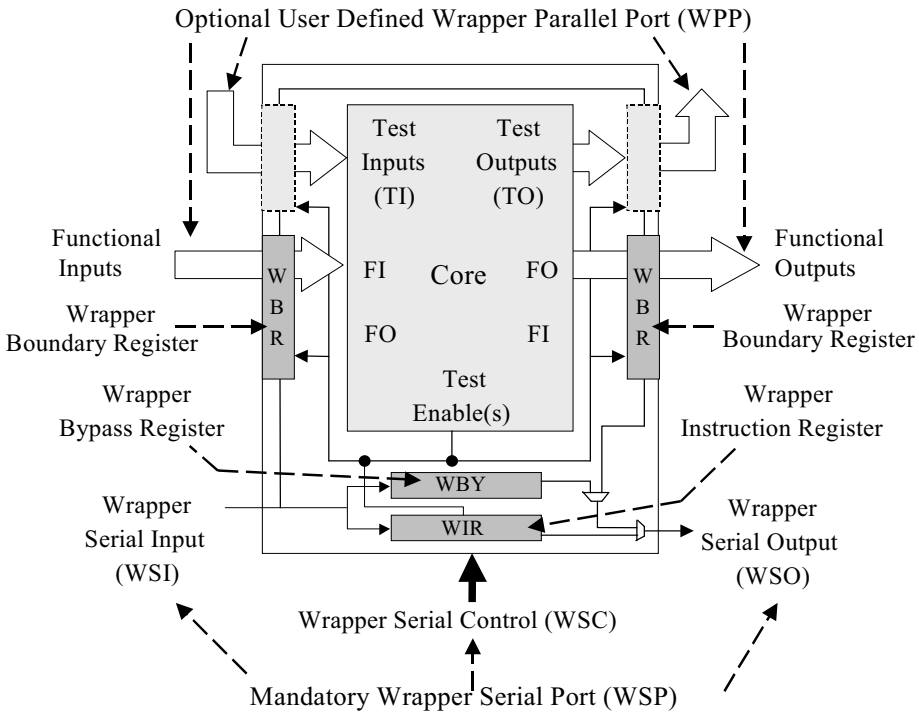


Figure 1.1: IEEE 1500 Standard wrapper structure [15].

1.3.1 Increasing Device Complexity and Test Data Volume

Thanks to the higher integration capability, more cores and blocks can be built in one chip to solve complex problems. SoC and System in Package (SiP) technologies are the typical examples. New DFT technology should be developed to conduct tests for these devices efficiently. One challenge is to supply controllability and observability for the embedded cores. The IEEE Standard 1500 [15] defines design methods to solve such problems. This standard was developed from the boundary scan IEEE 1149 Standard, which can be modified to adapt to multi-core testing. The IEEE 1500 Standard allows testing of individual cores on a whole chip, which is achieved by inserting a test wrapper for each core. The IEEE 1500 Standard wrapper is shown in Figure 1.1. It adds to each core independent instruction registers, boundary scan cells and bypass registers, so that we can conduct any test procedure on any core without affecting other parts of the chip. This greatly improves the flexibility in test planing.

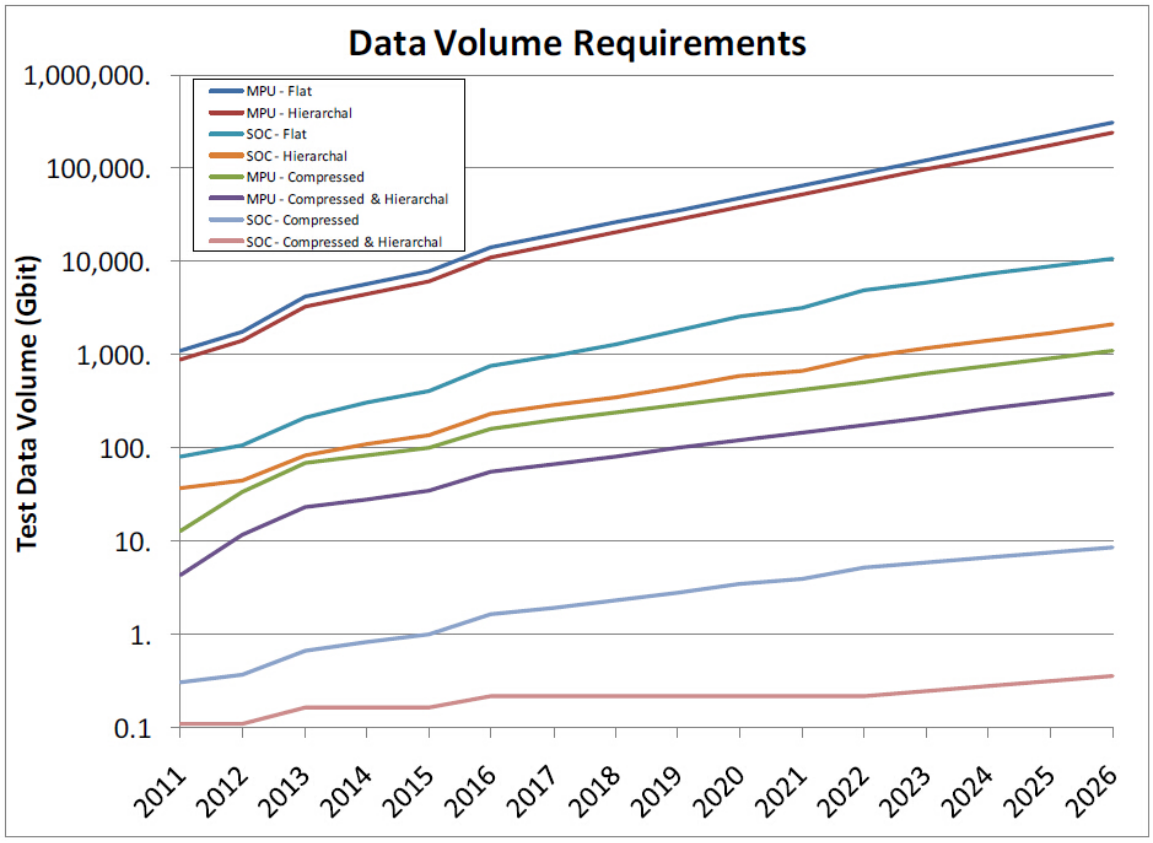


Figure 1.2: Test data volume trend with DFT techniques applied from [16].

Another concern is the huge amount of test data. It is one of the consequences of the circuit becoming enormously large. As the number of primary inputs (PIs) and registers increases, each test vector contains more bits. For a large design, the test data size can be on the order of Giga bits or even higher. A test data volume trend diagram from ITRS 2011 is shown in Figure 1.2.

In Figure 1.2, we can see that the test data volume for SoC devices that use the original test set can reach 10,000Gb in ten years. It also shows that some related technologies (test compression and hierarchical test) can effectively reduce the test data volume. However, for the multi-core processing unit (MPU) design, even with test reduction technologies, the test volume will be on the order of 100Gb by 2015. As a result, more techniques on test data manipulation should be investigated.

1.3.2 New Integration Technology: 3-Dimensional Stacked IC

As the feature size approaches close to sub-nanometer, we're about to hit the boundary where quantum effects begin to influence desired device properties. In order to reap the benefits of integrating more components in tiny areas as described by Moore's Law [59, 60, 61], chips are being stacked vertically. This technology is called 3-dimensional integration [47, 72]. The benefits of 3D stacked devices include the following:

- Yield: Instead of expanding die area, 3D technology stacks multiple small dies to achieve the same function. If unclustered defects are considered, the yield [24] of each die is:

$$Y = e^{-Ad}, \tag{1.1}$$

  in which A is the die area and d is defect density (average defects number per unit area). Clearly, larger die area ends up with lower yield. So, by 3D stacking, the total yield of the products can be improved.

- Performance: Circuit speed can also benefit from 3D stacking. In a large area single-chip (2D) device, some signal paths may need to be long, to cross the chip, resulting in large capacitance and resistance and therefore long path delay. In a 3D stacked device, such signals can travel through vertical path so that the length is greatly reduced, which leads to reduced path delay. By this method, the performance improvement is one of the 3D stacked device advantages.

- Power: Similar to performance comparison, the shorter signal and power supply paths dissipate less power as heat. Thus, with 3-dimensional routing, the power consumption is lowered.

- Heterogenous Integration: 3D stacked device can have different substrates or fabrication processes for different dies. This flexibility can never be gained by 2D devices because of structural and manufacturing limitations.

Besides benefits, 3D integration also brings problems, especially for test engineers. First, a new structure, through silicon via (TSV), is used for wiring between layers, whose reliability should be guaranteed by test [31]. Second, new test steps are needed due to new fabrication flow in producing 3D devices, such as pre-bond test, post-bond test, die-stack test [56], etc. Also, the test flows of 3D devices are different for different fabrication methods as there are wafer-to-wafer, die-to-wafer and die-to-die stacking methods [102]. Last but not the least, such complexity in test flow of 3D devices needs optimum test scheduling for test economics optimization [70, 71].

1.3.3 High Speed and Non-binary Interface Testing

Electronic devices can operate at multi-GHz frequency today. But normally we do not conduct functional test (for example stuck-at test) on DUTs at such high speed, because of overheating and other problems due to massive switching activities during test. However, we still want the high speed I/Os to be tested at the operating speed. The need to test high speed I/Os requires high speed ATE with ability to do complex measurements, such as jitter tolerance and generation according to [42]. Contrary to functional test, we can not test high speed I/Os with low speed ATE. Additional work must be done to increase the ATE speed. One approach is to use active test fixture in [63], which multiplies the ATE data speed.

Moreover, lots of communication devices (optical transceivers and wireless transceivers) don't use binary signal interfaces. So, in order to gain testability of non-binary interface, another direction is currently under exploration, finding ways to test other

non-binary interfaces like pulse amplitude modulation (PAM) and quadrature amplitude modulation (QAM) with digital tester. Reported work [38, 39] gives some solutions but here is still a long way to go in this area.

1.4 Test Scheduling and Test Resource Optimization

A big portion of test cost comes from the expensive ATE and the cost to run and maintain it. If the test resources are not fully utilized, the test cost per DUT increases. So it's better to have test resources fully exploited. Multi-site test [97, 98], concurrent test [68] and multi-domain test [88] are some of technologies targeting at test scheduling and test resources optimization. Optimizing the existing test resources brings down the test cost without too many changes on the ATE system. For example, the multi-site test tries to test as many DUTs as possible in parallel [21, 36, 40, 93, 94], which divides the test cost per DUT. To achieve better parallelism in multi-site test, RPCT technology is proposed. In current implemented RPCT [27, 30, 62, 76, 99], the TAM bandwidth is reduced by sending serialized test data. Also, the use of RPCT helps achieve bandwidth matching between ATE and DUT, whose benefit has been well discussed in [97, 98].

1.5 Test Compression Technology

As mentioned previously, growing test data volume is an emerging problem to be solved. Test compression technology is developed to reduce the volume of test data, so that the storage requirement on tester and test application time can be reduced. A survey [91] of test data compression technology is conducted years ago. Back in 1999, there was a test data compression algorithm [41] based on statistical coding. Later, there were more compression techniques being proposed in literatures based on coding algorithms [23, 28, 29, 43, 44, 53, 77, 89, 90, 96], such as optimal selective

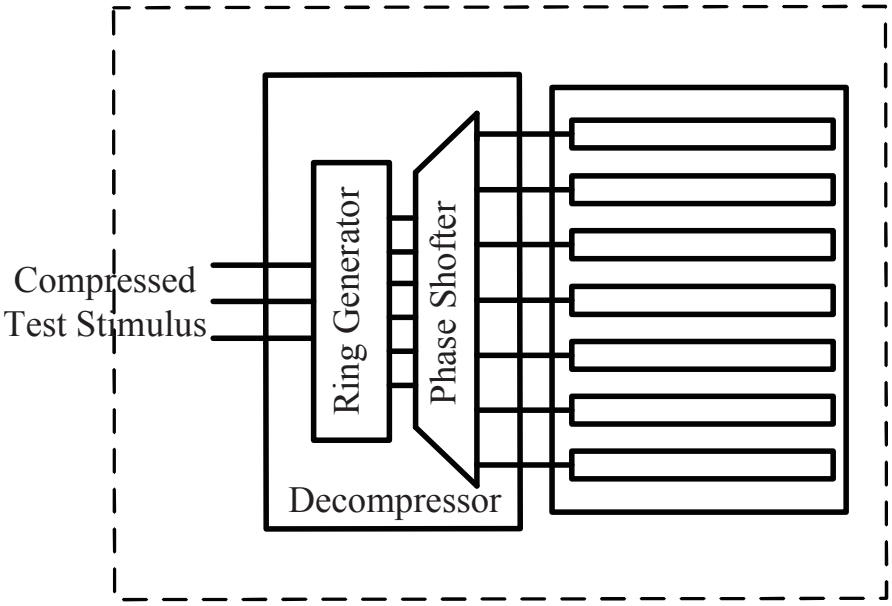


Figure 1.3: Decompressor structure of EDT compression.

Huffman coding [43] and bitmask-based code compression [77]. Although these code-based methods can cut off about 50% to 80% of test volume, this performance is not good enough for the rapid growing test volume. Together with other drawbacks like large area overhead, they are never implemented in industry.

On the other side, other mature compression technologies can be categorized into two types: linear-decompression-based scheme and broadcast-scan-based scheme [100]. They have been able to compress test data volume by 100x, such as adaptive scan [82, 84] in Dftmax [13] from Synopsys, embedded deterministic test (EDT) [33, 74, 75] in Testkompress [11] from Mentor and OPMISR+ [22, 45, 46] in Encounter DFT Architecture [6] from Cadence. These tools are based on lossy compression algorithms.

Next, I'll give an illustration of how test compression technology reduces test volume. Here the Testkompress compression method is shown as the example. The core idea of Testkompress is called EDT, which is proposed in [75]. In hardware, the decompressor of basic EDT is made of a ring generator and a phase shifter, as Figure 1.3 shows.

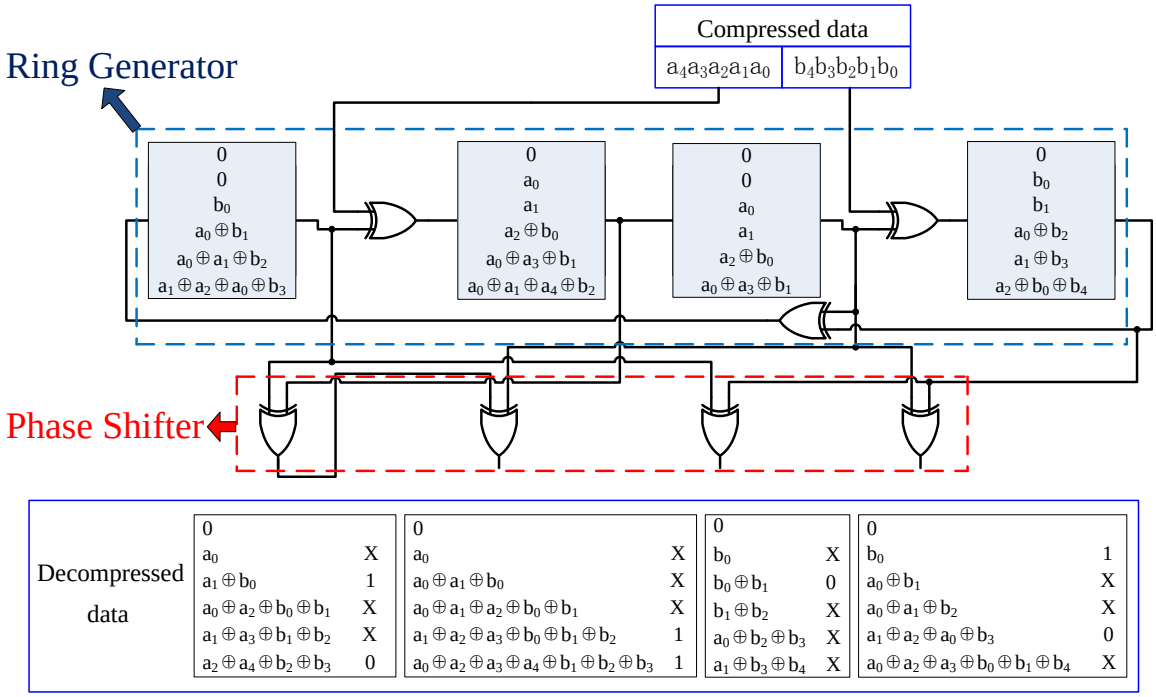


Figure 1.4: Test data decompression of EDT.

As described in [66], the ring generator is a multiple-input signature register (MISR) with primitive polynomial. The phase shifter is made up of an exclusive-OR (XOR) tree. Figure 1.4 shows how EDT compresses and decompressed test data.

We can see that the ring generator is set to all 0s in the beginning. For each cycle, the ring generator reads the compressed test data and produces some value in the linear feedback shift registers (LFSR). Those values will go through the phase shifter and become decompressed test data, which can be presented in terms of compressed test data. Then for those decompressed data that need to be set to certain values (1 or 0), we make equations. Ignoring the don't-care bits, we can have a set of equations whose variables are the compressed test data. The equation set for this example is Equation 1.2, as given next.

$$
\begin{aligned}
&a_1 \oplus b_0 = 1; \quad a_2 \oplus a_4 \oplus b_2 \oplus b_3 = 0; \\
&a_1 \oplus a_2 \oplus a_3 \oplus b_0 \oplus b_1 \oplus b_2 = 1; \\
&a_0 \oplus a_2 \oplus a_3 \oplus a_4 \oplus b_1 \oplus b_2 \oplus b_3 = 1; \\
&b_0 \oplus b_1 = 0; \quad b_0 = 1; \\
&a_1 \oplus a_2 \oplus a_0 \oplus b_3 = 0.
\end{aligned} \tag{1.2}
$$

Then solving the equation set, we'll get the compressed test data which should be stored in the ATE. The decompressor can recover the original test from the decompressed test data. This is just the fundamentals of how EDT works. To maximize the compression efficiency, lots of other tricks are applied in the actual Testkompress scheme, which are not shown here.

1.6 RPCT Technology in Testing

Optimizing testing resources is an important consideration to save testing cost. For a long time, the concept of multi-site test helped increase the test throughput and reduce individual test cost for higher parallelism. RPCT is quite a support for improving test parallelism. The benefit of RPCT was discussed in [21, 36, 40] for multi-site test. Currently, the way to implement RPCT is based on test data serialization and deserialization. Here I use the term serializer/deserializer (SerDes) to stand for this technique. It only sends serialized data in normal speed. In [62] and [76], SerDes is used for time-multiplexing the test channels for reutilization. In [30], Synopsys enables pin-limited mode in their test compression scheme, using SerDes. Moreover, a supplementary work [27] to EDT compression adopts SerDes as well. Figure 1.5 shows the traditional way to apply test data, in which multiple test channels feed into equal number of scan chains.

Figure 1.6 shows how test data is applied in RPCT with SerDes technology, in which a single test channel feeds test data into multiple scan chains serially. The test

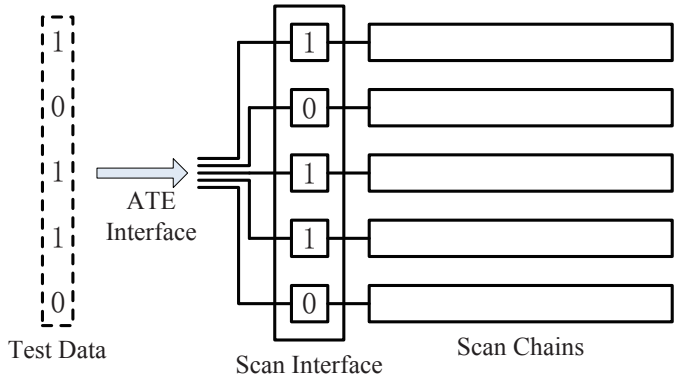


Figure 1.5: Traditional test application method.

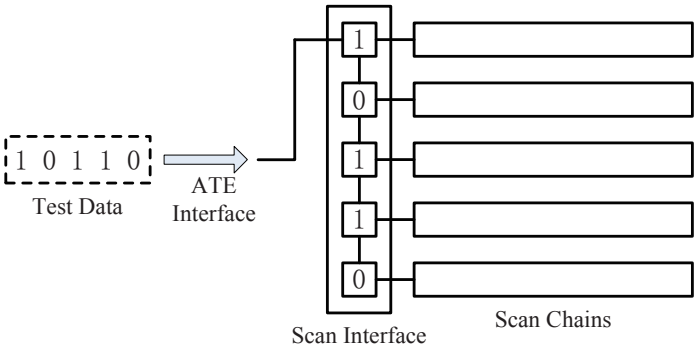


Figure 1.6: Test application method in RPCT.

data are serialized in the ATE and deserialized in DUTs. Such a scheme provides a way to match the bandwidth between ATE and DUTs as discussed in [27, 98], etc. Moreover, this implementation doesn't need any modification on the ATE and the modification on the DUT is quite simple and cheap. That's why this scheme has become the most popular implementation for RPCT.

However, when RPCT comes across with test compression, the data rate limit of the test channel will take effect to limit the test speed. In [27], the author drew the conclusion that the number of inputs of each decompressor interface should be above a certain quantity to break the correlation barrier, so that the volume of the top-off patterns to compensate the lost fault coverage (FC) can be reduced. Figure 1.7 shows the change in the RPCT interface as the device trend goes to multi-core SOC from

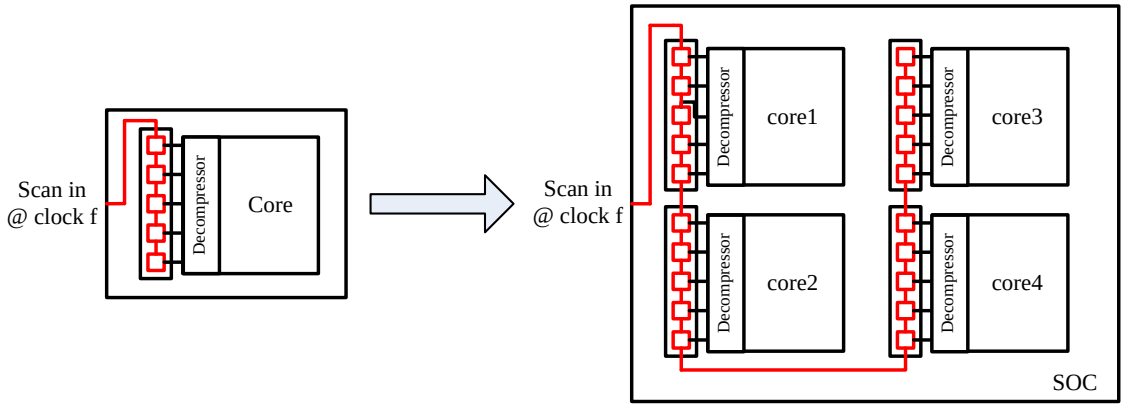


Figure 1.7: RPCT on single-core and multi-core design.

single-core device. In a single core design, there is just one RPCT interface. In this case, the scan speed is one fifth of the clock frequency. For multi-core SoCs, multiple RPCT interfaces are concatenated. In this case, the scan speed is one twentieth of the clock frequency, which dramatically slows down. In conclusion, because the length of RPCT interface can be quite long with multi-core SoCs, and in order to match the bandwidth between ATE and DUT, the scan speed should equal the test channel data rate divided by the length of the shifter registers in the deserializer, resulting in rather slow scan speed for SoCs. The problem of slow scan speed will be worse as the complexity of SoCs increases.

1.7 The Objective of This Thesis

In multi-site test, RPCT technology benefits by increasing parallelism, but also causes problems like test speed reduction, as discussed in previous section. The conflict arises from the limited data rate with few test pins and growing width of the deserializer interface. This work explores a new method to apply test data so that the data rate per test channel can be improved. With higher data rate on a test channel, the above problem in RPCT can be greatly relieved and furthermore, such technology can also be the universal solution for test time reduction in general IC testing.

Chapter 2

Increasing the Data Rate of Test Channel

The importance of increasing test channel data rate has been well discussed. Next, I will show some possible options. For any kinds of data link, the data rate (bit rate R) can be expressed as:

$$R = f_s \cdot \log_2 N, \tag{2.1}$$

where f_s is the Baud rate and N is the total number of different codes in a symbol. For the commonly used binary links, f_s is the clock frequency and N equals 2. It's obvious that the only way to boost data rate for binary links is to increase the clock frequency f_s. But in other communication systems, people use different formats of signals, some of them can increase the data rate by adding more codes in one symbol (increasing N).

2.1 Increasing Clock Frequency

As stated, the data rate of certain data link is determined by its clock frequency. SerDes is a widely used technology to achieve high speed data link, which has been adopted in fields like ethernet transceivers [17], peripheral component interconnect (PCI) Express [25], serial advanced technology attachment (SATA) [18] and field programmable gate arrays (FPGAs) [1], etc. It replaces multiple parallel links by a serial link with a boosted clock, which guarantees the data rate of the serial link equals to the total data rate of the parallel links, which is shown in Figure 2.1. The advantages of SerDes technology are described in [19], such as maximum data flow,

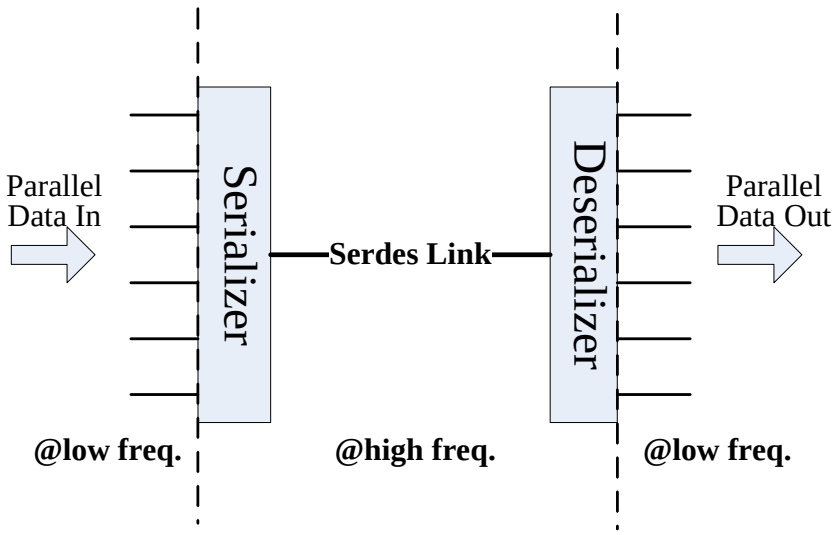


Figure 2.1: Conceptual diagram of SerDes data link.

low pin count, less simultaneous switching outputs (SSO) problem [79], etc. Some academic explorations in adopting such technology in ATE systems can be found in [55] and [92]. In these references, the author tries to build high speed test channels by using the SerDes port in FPGA. However, this method has a couple of drawbacks for massive product testing, such as: the large power consumption for high speed switching activities in the accelerated binary signal; and the high frequency bandwidth requirement on the link wire for a big portion of high frequency parts in frequency response. If such test interface is integrated with all test channels, the test fixture design and signal wiring can be very difficult and costly [64]. Therefore a better way should be found to speed up the test channel. Next, I will introduce another way to increase data rates of data links.

2.2 Increasing Information per Symbol

By adding more codes (N) in each symbol, some digital modulation techniques such as phase-shift keying (PSK), QAM and PAM [35], etc. are developed to achieve

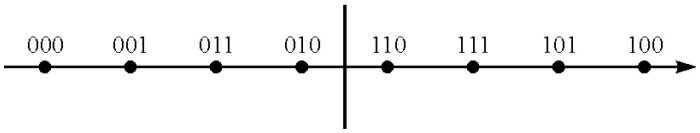


Figure 2.2: Constellation diagram of 8-PAM modulation.

higher data rate. These techniques have the advantage of reducing frequency bandwidth requirement as well. The M-lvl PAM bandpass signal with increased symbol sequence for every T seconds can be expressed as:

$$s(t) = a(n)p(t - nT), \tag{2.2}$$

where $p(t)$ is the unit pulse function. The constellation diagram of 8-PAM signal is shown in Figure 2.2. From this figure, we can see that there are 8 codes to convey 3 bits of binary information. The baseband PAM is used in implementing ethernet, optical and serial link transceivers [69, 83], etc.

The baseband PAM signal is actually a multi-values logic signal, referred to as MVL signal for short in this dissertation. As the name indicates, MVL signal has multiple (> 2) voltage levels within one symbol so that the information of each symbol is more than a single bit. Study on MVL signal and circuit has become a field of research. For example, the MVL signal can be used to reduce circuit area by using compact computing elements [58, 80, 81], to enhance memory storage by saving multiple-bit information with a single memory cell [48] or to reduce the power consumption of data bus [86, 87], etc. Some new idea [85] is proposed to use new transistor structure for MVL implementation in the future.

To further increase the number of codes in each symbol, QAM is proposed, for which the M-lvl QAM bandpass signal can be expressed as:

$$s_i(t) = A_i \cdot p(t) \cdot \cos(2\pi f_0 t + \phi_i) \qquad i = 1, 2, ..., M, \tag{2.3}$$

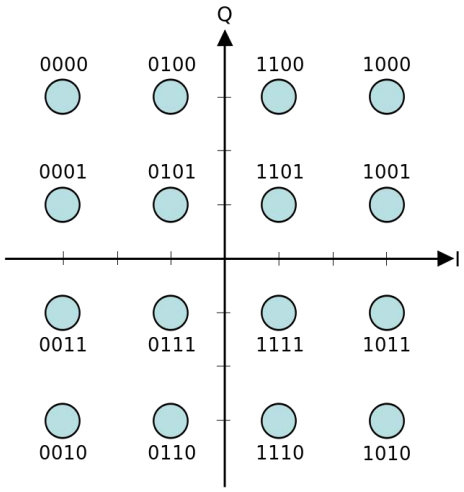


Figure 2.3: Constellation diagram of 16-QAM modulation.

where A_i is the amplitude, ϕ_i is the phase and $p(t)$ is pulse function. If we express A_i and ϕ_i by orthogonal decomposition, they become:

$$\begin{aligned} A_i &= \sqrt{A_i^{I2} + A_i^{Q2}} \\ \phi_i &= \tan^{-1}(\frac{A_i^I}{A_i^Q}) \qquad i = 1, 2, ..., M. \end{aligned} \tag{2.4}$$

With quadrature components (I and Q), QAM bandpass signal can be expressed as:

$$s_i(t) = A_i^I p(t)\cos(2\pi f_0 t) - A_i^Q p(t)\sin(2\pi f_0 t) \qquad i = 1, 2, ..., M, \tag{2.5}$$

A 16-QAM constellation diagram is shown in Figure 2.3. We can see that with different amplitudes and phases, there are 16 codes in one 16-QAM symbol, so that each 16-QAM symbol contains 4 bits of binary information. QAM is widely used in transceiver design [78, 101] too.

2.3 Choosing the Way to Increase Test Channel Data Rate

The advantage of modulation over increasing clock frequency has been discussed in previous section, which can both increase the data rate and reduce the frequency

bandwidth requirement. Further more, if we change the format of data on test channels, we have to consider about the difficulty to decode the new format of signal back into binary signal in DUT. Due to these considerations, we choose the MVL signal as the new test data format to boost the data rate for test channels, for its simple decoder structure.

Chapter 3

MVL Test Application

Although the ATE used in industry may not readily have the capacity to generate MVL signals, there has been some progress made toward this goal. In [39], Advantest proposed an ATE interface with capacity to generate 16 Gb/s 4-PAM signal for ethernet interface testing. They can even make ATE generate 64Gb/s 4-PAM signal in a later work [65].

Figure 3.1 shows a simple multi-valued logic test application structure. A multi-valued logic generator in the ATE converts multiple digital bits into single voltage level. A multi-valued logic decoder in the DUT converts the voltage level back into digital bits and feeds them into test structures (like multiple scan chains shown in Figure 3.1). In order to perform the conversion from binary bits to multi-valued signals, we use an N-bit DAC. On the other end, we use an N-bit ADC to do on-chip multi-valued to binary decoding. Compared to traditional ATE, the only modification is on scan-in path: a DAC is added and the test data are sorted into N-bit slices, accordingly.

On the ATE side, an ADC is inserted as the multi-valued logic decoder. The decoded pattern is going to be fed into scan chains or decompressor in a compression scheme.

3.1 MVL Test Application in Multi-site Test of SoCs

The MVL test application method needs modifications on both ATE and DUTs. Modifying an ATE system seems expensive but feasible, adding only to the fixed cost. Actually, the hardest problem comes with the modification of DUTs, because

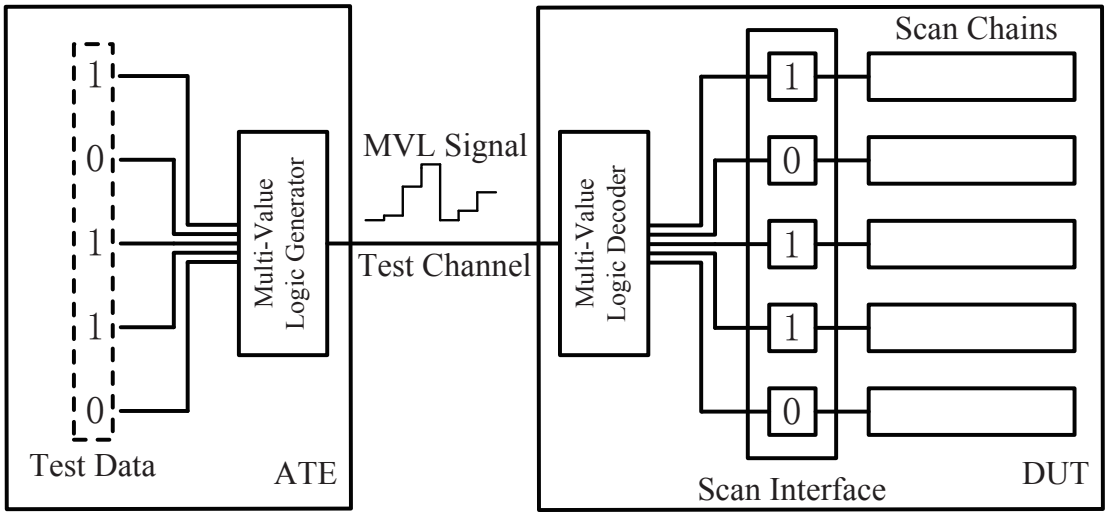


Figure 3.1: Overview of multi-value logic test application method.

we cannot afford multiple ADCs being integrated in DUT with current data converter technology, which will contribute to high cost for each DUT. However, it still has a promising application in a popular test scheme. In multi-site test, people always want a DUT to take less tester resources for testing. So it has been one of the mainstreams to keep DUTs with very narrow test access mechanism (TAM) in multi-site test, which is known as RPCT technology. For the DUTs that support RPCT, the width of TAM can be minimum: one-pin TAM. In the literature [27, 62, 76] describing such technology, the so-called RPCT technology is adopted, which has a single pin to receive serialized test data. For these RPCT based designs, only one ADC needs to be integrated in the DUT, making it practical with current data converter technology. This is one major reason to apply MVL test application on RPCT devices. But still, as data converter technology develops, the MVL test application can be used on general purpose testing when it becomes acceptable to integrate multiple ADCs in a DUT.

3.2 Selecting Components for MVL Test Application

In this section, we discuss the requirements of DAC/ADC and give some candidate solutions.

- Speed consideration:

  The ATE speed generally refers to the clock or data rate it supports. According to different test schemes, slow ATEs (10-50MHz) or fast ATE (up to 500MHz) will be set in use, which means the conversion speed of DAC/ADC should follow the ATEs. Decades ago, flash ADC is the only ADC structure which can meet fast ATE speed requirement. More recently, as the data converter technology has developed, we have some other options. From an ADC survey [20] from Murmann, there are more ADC types reported to be capable of handling multi-hundred MHz input.

- Space, power and cost consideration:

  As the MVL signal generator, a DAC is integrated with the ATE, so it almost has free space and power budget. Moreover, the DAC only adds to fixed cost which is ignorable in large volume devices testing. So we are concerned less about the DAC in power consumption and area overhead. On the other hand, we have more strict constraints on ADCs, since they are integrated on-chip. To handle the strict power consumption and area overhead budget, the ADC structure should be carefully picked, and the ADC resolution cannot be very high for it exponentially increases power consumption and area overhead.

- Noise consideration:

  Because analog parts are involved in this scheme, the impact of noise becomes more crucial. For MVL signals, the noise margin is cut by half for every bit

added in data converter resolution. This becomes another restriction on data converter resolution.

As discussed, choosing a DAC for MVL generator is comparatively easier. Some reported DACs like [32] and [73] are good enough for high speed ATE application, and are based on current-steering structure. On the other hand, choosing a high speed ADC for a MVL decoder is much more difficult, but with the advanced ADC technology discussed in Murmann's tutorial [67], we can still find the best solution: Time-Interleaved SAR ADC. One of this type [49] succeeded in solving area, power and speed issues concurrently. For fast ATE cases, we can choose current-steering DAC in the ATE and SAR ADC in DUT. For those slower ATE cases, we have more flexibilities in choosing data converter types. There is a trade-off in picking data converter resolution also. Concerns about power consumption, area overhead and noise margin prevent high resolution data converters. But the data converter resolution determines the improvement in data rate. So for the objective of increasing data rate, we want the resolution as high as possible. In the following chapters, we'll discuss about how resolution affects testing reliability.

Chapter 4

Reliability Issues in MVL Test Application

In general, analog devices have reliability issues like noise, interference or process and environmental variations, which can lead to malfunction. We proposed the idea in [51] that two major reliability problems exists for MVL test application. First, data converters have nonliearities. The output voltage of DAC and judge ranges of ADC can not be ideal, which may cause permanent misinterpretation in design-to-be-matched converter pairs. We name such situations as *intrinsic failure*. Second, noise problem becomes critical because of the shrunk noise margin. For an N-bit converter pair, the bit-level noise margin is divided by 2^N, compared to the binary signal. If noise causes false decoding of sent test data, we call it an *noise-induced application error*. In this chapter, all the mathematical analysis is based on Matlab [57] program.

4.1 Mathematical Analysis of Intrinsic Failure

In order to bring nonlinearities into consideration, we assume the DAC output voltage of each code is Gaussian distributed at its ideal position instead of a set value. Similarly, the reference voltages of ADC have Gaussian distribution too. This way of nonlinearity modeling is used in [54]. Figure 4.1 shows a Gaussian distributed DAC output and its corresponding ADC judge range which is set by two Gaussian distributed reference voltages. There are two overspread areas for DAC output and ADC reference voltage distributions. Therefore, it is possible that $Vout(n)$ of DAC is larger than $Vref(n)$ of ADC, in which case an originally sent "n" is interpreted as "$n+1$" by the ADC; or $Vout(n)$ may be smaller than $Vref(n-1)$ of ADC, in which case an originally sent "n" is interpreted as "$n-1$" by the ADC. Normally, INL

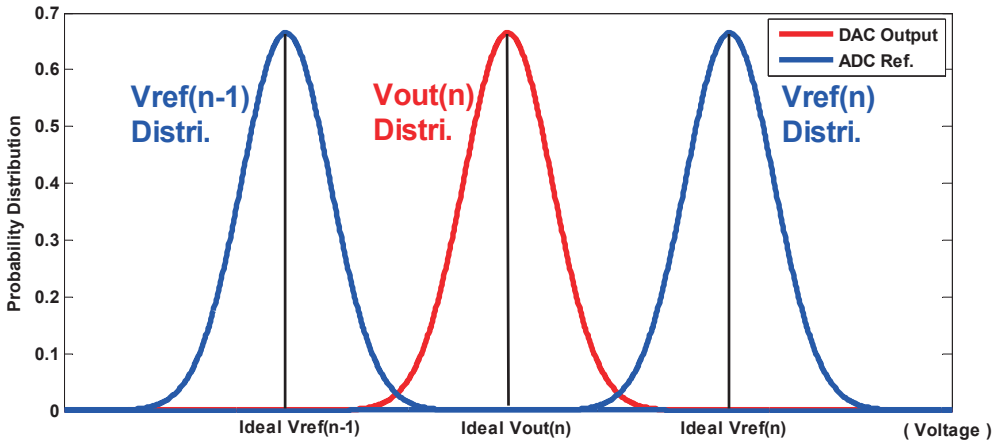


Figure 4.1: Nonideal DAC and ADC characteristics.

(Integral Nonlinearity), DNL (Differential Nonlinearity), full scale error and offset error are specifications to describe these deviations [37], among which INL and DNL are of the most concern. Both INL and DNL are scaled by LSB (Least Significant Bit). The definition of DAC LSB is:

$$LSB_{DAC} = \frac{V_{fs\_ideal} + E_{fs}}{2^N - 1}, \tag{4.1}$$

where, V_{fs_ideal} is the full-scale voltage in the ideal situation and E_{fs} is the full-scale error. The definition of ADC LSB is:

$$LSB_{DAC} = \frac{V_{ref\_ideal}(2^N - 1) - V_{ref\_ideal}(1)}{2^N - 2}, \tag{4.2}$$

where, $V_{ref\_ideal}(2^N - 1)$ is the largest reference voltage and $V_{ref\_ideal}(1)$ is the smallest reference voltage. For a given LSB, the definition of DAC DNL is:

$$DNL_{DAC}(n) = \frac{V(n+1) - V(n) - LSB_{DAC}}{LSB_{DAC}}, \tag{4.3}$$

where, $V(n)$ is the DAC output for code "n".

Table 4.1: DNL of N-bit flash DAC.

Code	0	1	2	...	2^N-3	2^N-2	2^N-1
Range	$-\infty$ $ref(0)$	$ref(0)$ $ref(1)$	$ref(1)$ $ref(2)$	...	$ref(2^N-4)$ $ref(2^N-3)$	$ref(2^N-3)$ $ref(2^N-2)$	$ref(2^N-2)$ $+\infty$
DNL	–	$DNL(1)$	$DNL(2)$	...	$DNL(2^N-3)$	$DNL(2^N-2)$	–

We should point out that the definition of ADC DNL varies across the literature. Because only 2^N-1 comparators are needed to build an N-bit flash ADC, codes 0 and 2^N-1 coincide with side boundaries, so no definitions of DNL are needed for these two codes, as shown in Table 4.1. Here we use the following equation to represent ADC DNL, specifically for flash ADC:

$$DNL_{ADC}(n) = \frac{V_{ref}(n) - V_{ref}(n-1) - LSB_{ADC}}{LSB_{ADC}}. \tag{4.4}$$

Next, we express $V(n)$ in terms of DNL so that the nonliear effects are considered:

$$V(n) = V(0) + [n + \sum_{1}^{n-1} DNL(n)] \cdot LSB_{DAC}, \tag{4.5}$$

in which $n \in \{1, 2^N-1\}$ and $V(0) = V(0)_{ideal} + E_{offset}$.

In order to have correctly working converter pairs, the output $V(n)$ of DAC should be set in the range of the nth code of ADC, which is described as following:

$$\begin{aligned} &V(n) \in \{V_{ref}(n-1), V_{ref}(n)\},\ n \in \{1, 2^N-2\}; \\ &V(0) \in \{-\infty, V_{ref}(0)\}; \\ &V(2^N-1) \in \{V_{ref}(2^N-2), +\infty\}. \end{aligned} \tag{4.6}$$

If the data converter pair violates the above rules, we mark them as intrinsic failures. This undesirable situation will lead to malfunctioning data converter pairs. Such a

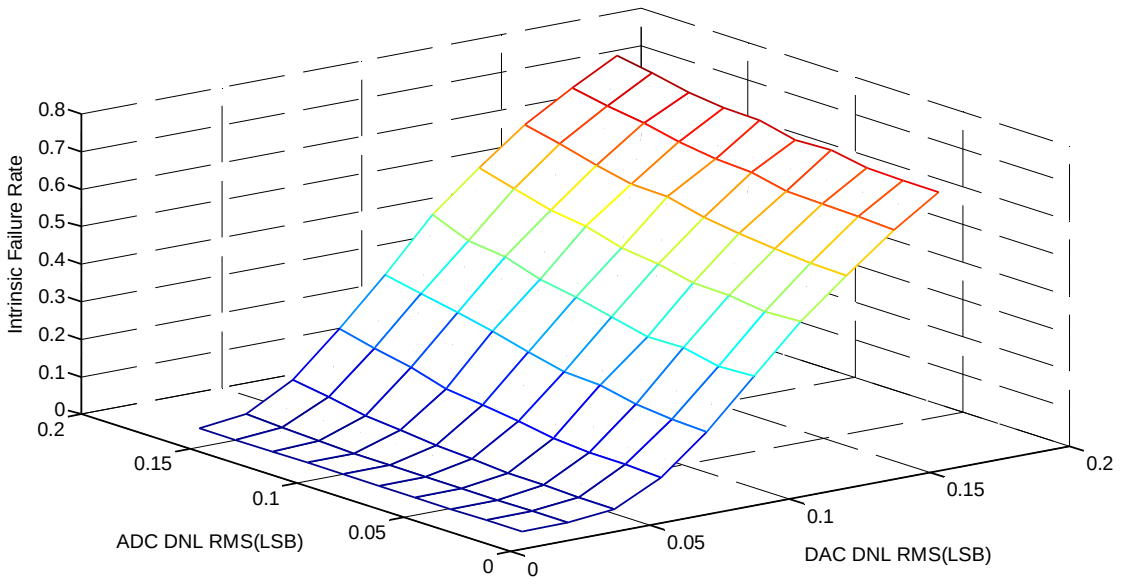


Figure 4.2: Intrinsic failure rate vs. data converter nonlinearities.

failed converter pair will always falsely send a certain code, so that it cannot be used for MVL communication. Next, we need to observe how the scale of data converter nonlinearities affects the intrinsic failure rate. In a simulation, we randomly generate 10000 4-bit nonlinear data converter pairs as one group (with the same variance for the Gaussian distribution), for which the RMS DNL and the number of intrinsic failed pairs are recorded. By varying the variance, we get 100 groups with different DNL from 0 LSB to 0.15 LSB, as shown in Figure 4.2.

From the simulation result, we can see that the number of failed pairs is increasing as the DNL of ADC/DAC increases. Also, we find that when RMS DNL of DAC is the smallest (0.0166LSB), the failed number increases from 0 to 11 across the whole ADC DNL range. When RMS DNL of ADC is the smallest (0.0166LSB), the failed number increases from 0 to 6946 across the DAC DNL range. It shows that the intrinsic failure rate is primarily determined by the DNL of DAC. So, in order to have fewer intrinsic failures, the performance of DAC is more important than ADC. In manufacturing testing, there is only one ATE testing thousands of chips. The performance of the DAC in ATE can be relatively high because we can select a

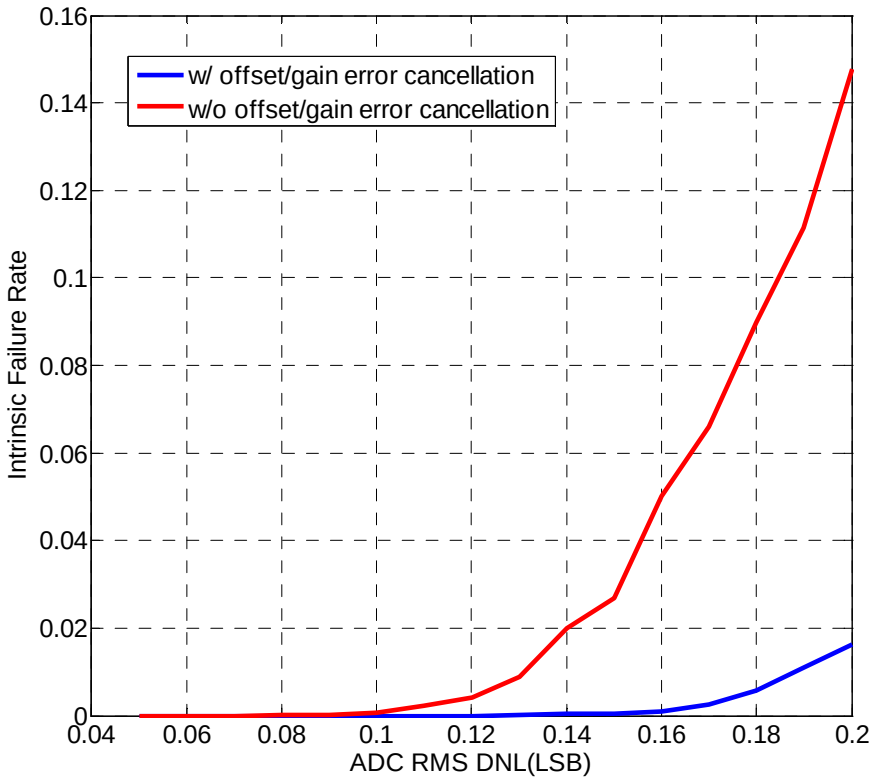


Figure 4.3: Intrinsic failure rate vs. ADC DNL with and without offset/gain error cancellation.

DAC with desired performance and apply many correction techniques to ensure high linearity. However, the ADCs should be small and cheap to be integrated on every chip. So in the next step, we assume that the ADCs have basic flash structure with worse performance than the DAC in ATE.

Another model is simulated with manufacturing testing conditions. In this model, we have one DAC converter in the ATE and 1000 ADC decompressors in DUT as a test case. In total, there are 100 such test cases. For the DAC, we assume that it has the RMS DNL as small as 0.05LSB. To show how ADC nonlinearities affect the failure rate, we increase the RMS DNL of ADC from 0.05LSB to 0.2LSB. We separately simulate the test case with DAC to verify whether the offset and gain error is canceled. As Figure 4.3 shows, the DAC without offset and gain error cancellation has an average number of intrinsic failed chips as 147.6 per 1000 at 0.2LSB ADC

RMS DNL. For the DAC with offset and gain error cancellation, the average number of intrinsic failed chips is only 16.3 per 1000. The simulation result shows that the case with calibrated DAC has much fewer intrinsically failed parts.

4.2 Mathematic Model of Noise-Induced Error

Next, I will analyze how noise affects data conversion. To simplify the mathematical model, we generalize a white noise to add on test channel, standing for all kinds of noise which the test application system may suffer, including DAC and ADC noise, environmental interference, and on-chip digital switching noise. We use parameter symbol error rate (SER) in determining the system reliability in the presence of noise. Assuming the ideal situation, data converters have perfect transfer functions so that every DAC output code's level sits in the middle of the corresponding ADC judge range, in such a way that it maximizes the noise margin. The *SER* in ideal case can be calculated as: $erf(\frac{0.5\Delta V}{\sqrt{2\sigma^2}})$, in which ΔV equals to $\frac{V}{2^N}$ and N is the data converter resolution. Because increasing resolution of data converters shrinks the noise margin, we can see that *SER* increases with data converter resolution rapidly in Figure 4.4.

If we bring in the nonlinearities, the noise margin should be further reduced, making *SER* even higher. In our simulation considering data converter nonlinearities, we generalize the condition for a good MVL decoding as:

$$\begin{aligned} &V_{ref}(n-1) < V_{out}(n) + V_{noise} < V_{ref}(n), \\ &V_{ref}(n-1) - V_{out}(n) < V_{noise} < V_{ref}(n) - V_{out}(n), \end{aligned} \tag{4.7}$$

in which $V_{ref}(n)$ is the nth reference voltage of ADC, and $V_{out}(n)$ is the output voltage of the nth code in DAC. V_{ref} and V_{out} are distorted because of nonlinearities. Equation 4.7 shows that if the DAC's output voltage of the nth code with noise added still locates in between $V_{ref}(n-1)$ and $V_{ref}(n)$ of ADC, the decoded signal is identical to DAC inputs. According to the definition of cumulative distribution

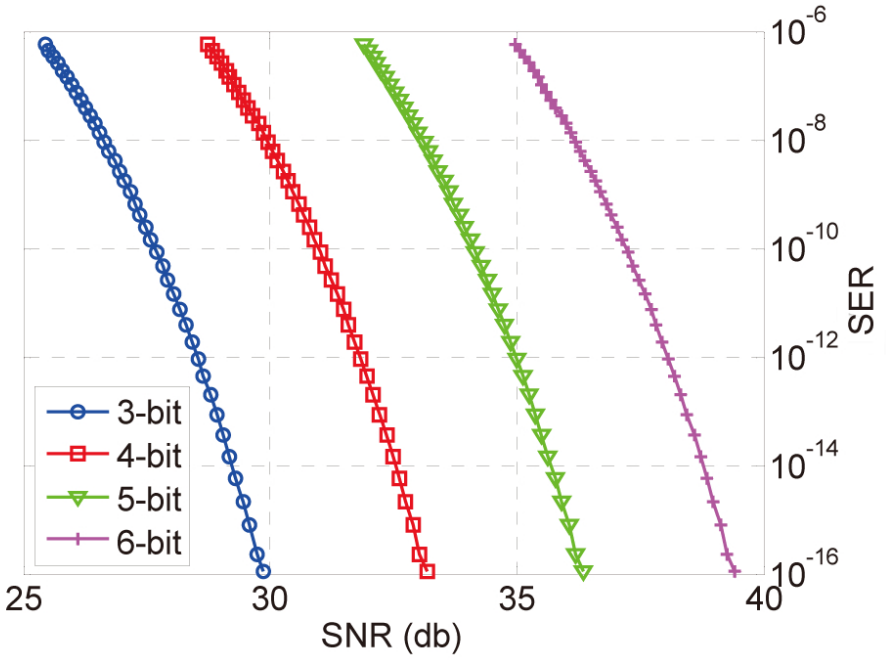


Figure 4.4: Ideal SER vs. SNR with different data converter resolution.

function (CDF), the probability for V_{noise} setting into the above region should be:

$$\begin{aligned} &P(V_{ref}(n-1)-V_{out}(n) < V_{noise} < V_{ref}(n) - V_{out}(n)) \\ &\qquad = CDF(V_{ref}(n) - V_{out}(n)) - CDF(V_{ref}(n-1) - V_{out}(n)); \end{aligned} \tag{4.8}$$

Then the correct decoding probability for the nth code should be:

$$\begin{aligned} P(n) =& CDF[V_{ref}(n) - V_{out}(n)] - CDF[V_{ref}(n-1) - V_{out}(n)] \\ =& 0.5\{1 + erf[\frac{V_{ref}(n) - V_{out}(n)}{\sqrt{2\sigma^2}}]\} - \\ & 0.5\{1 + erf[\frac{V_{ref}(n-1) - V_{out}(n)}{\sqrt{2\sigma^2}}]\} \\ =& 0.5\{erf[\frac{V_{ref}(n) - V_{out}(n)}{\sqrt{2\sigma^2}}] - erf[\frac{V_{ref}(n-1) - V_{out}(n)}{\sqrt{2\sigma^2}}]\}. \end{aligned} \tag{4.9}$$

Finally, we estimate the *SER* by averaging the correcting decoding probabilities of all codes:

$$SER = 1 - \frac{\sum_{n=0}^{2^{N_{wid}}-1} P(n)}{2^{N_{wid}}}, \tag{4.10}$$

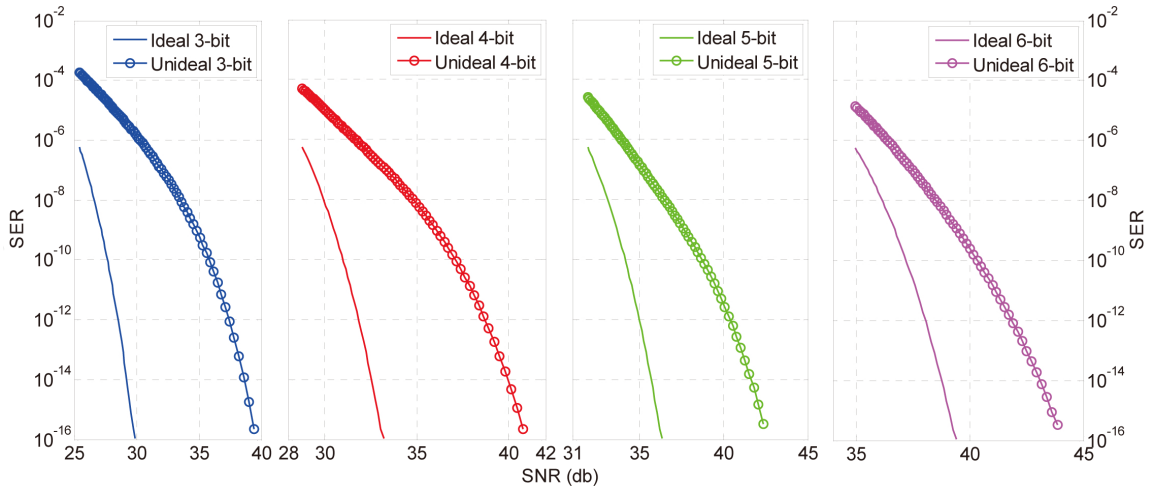


Figure 4.5: SER vs. SNR for nonlinear data converters with different resolutions.

where N_{wid} is the resolution of the MVL signal.

For statistical analysis, we make one group including one ATE and 1,0000 DUTs, and run 100 times for data gathering. We conduct such simulation with five different resolutions N_{wid}. There are three significant variables in a particular conversion: run time l, DUT m and code n, in which $l \in \{0, 99\}$, $m \in \{0, 9999\}$ and $n \in \{0, 2^{N_{wid}} - 1\}$. The average SER in our test application method is:

$$SER = 1 - \frac{\sum_{l=0}^{99} \sum_{m=0}^{9999} \frac{\sum_{n=0}^{2^{N_{wid}}-1} P(l,m,n)}{2^{N_{wid}}}}{10^7}; \tag{4.11}$$

Figure 4.5 shows the relation between SER and SNR (signal to noise ratio) with different converter resolutions, in which ADC/DAC DNLs are 0.2LSB and 0.1LSB, respectively. Compared with the ideal situation, the SER of converter pair becomes worse for nonlinearities. The 3/4/5/6-bit data converter pairs require 9.5/7.5/6.5/5 db more SNR at 10^{-15} SER. In our simulation, to reach an acceptable SER as 10^{-10}, The 3/4/5/6-bit data converter pairs require 36/37/38.5/40.2 db channel SNR, respectively.

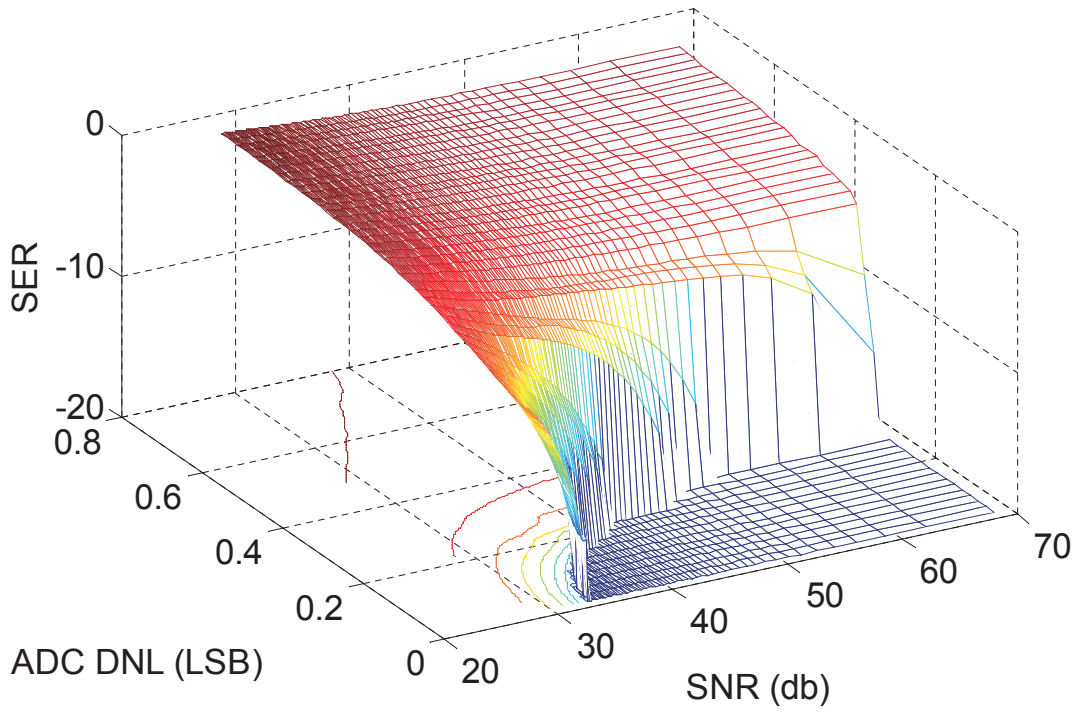


Figure 4.6: SER vs. ADC DNL and channel SNR.

Next, we keep DAC having 0.1LSB DNL and alter ADC DNL and channel SNR. The change of *SER* is shown in Figure 4.6. On the axis of ADC DNL, the *SER* increases as ADC DNL increases. When ADC DNL is around 0.4LSB, increasing channel SNR does not do much to improve *SER*.

Chapter 5

Assuring MVL Test Reliability

From the mathematical analysis of MVL communication reliability in the previous chapter, we find that the accuracy of data converters plays an important role in ensuring correct MVL communication. However, with current data converter technology, we have to compromise between accuracy, area and power. Therefore, a way to protect reliability with compromised data converter accuracy should be found. Two techniques corresponding to *intrinsic failure* and *noise-induced error* are proposed in [52] and further improved in [50]. For *intrinsic failure*, a calibration scheme cancels on-chip ADC nonlinear effect. For *noise-induced error*, an error detection and correction scheme detects erroneous test application and reapplies the test data.

5.1 Calibration of ADC Nonlinearity

It has been shown that both DAC and ADC nonlinearities contribute to the increase of intrinsic failure rate and noise margin reduction. Here I propose the idea to calibrate ADC nonlinearity.

5.1.1 Avoiding Failures from ADC Nonlinearity

Here we make an assumption that it is much easier to control the performance of DACs in ATE and, therefore, the major task is to calibrate ADC nonlinearity. We realize that after fabrication the performance of on-chip ADC is fixed and we lack the capability to adjust ADC at that point. In an uncalibrated case, the transfer function of DAC and ADC may look like what is shown in Figure 5.1. Because of nonlinearities, neither DAC's nor ADC's transfer function is linear. Then a mismatch

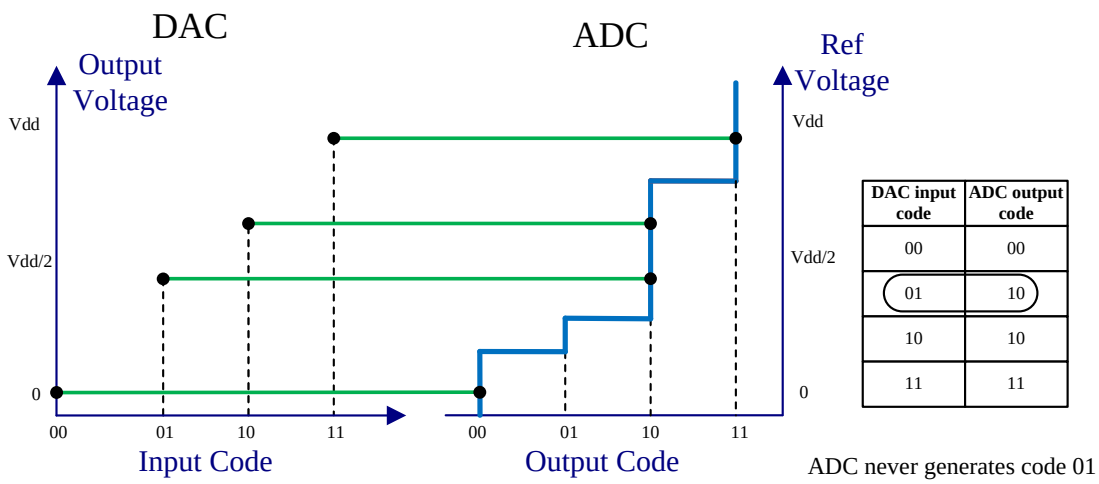


Figure 5.1: Transfer function of an uncalibrated 2-bit DAC and ADC pair.

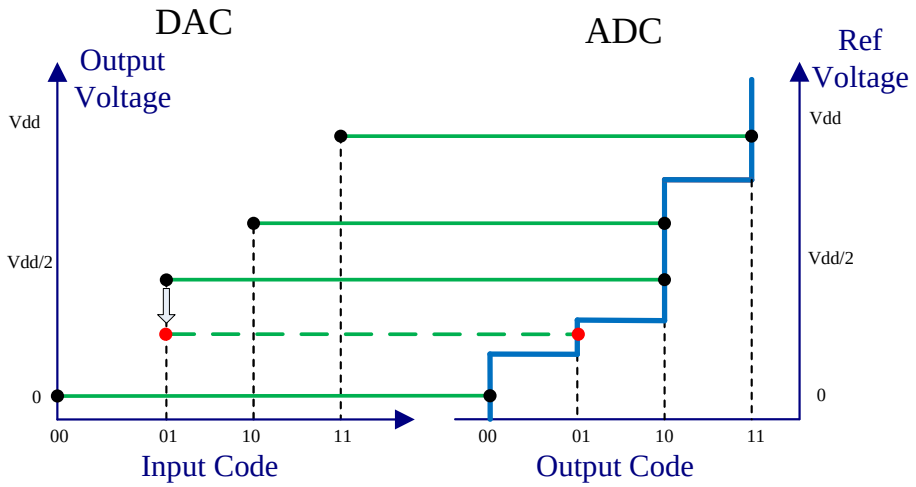
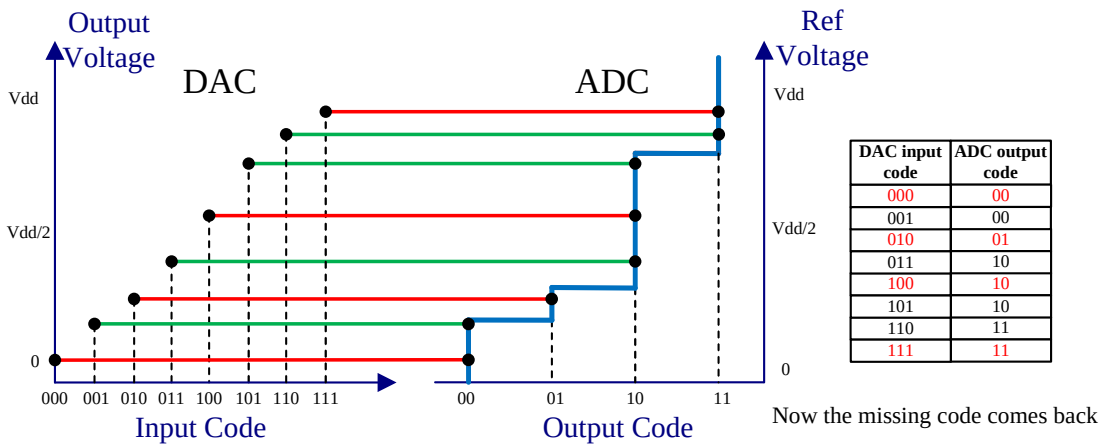


Figure 5.2: Preventing intrinsic failure by adjusting the DAC output.

exists for this data converter pair. This instance illustrates how the intrinsic failure happens: there is no chance for this ADC to produce code 01 with the DAC output, which is known as intrinsic failed. In such condition, we can only do something with the DAC to eliminate the error. If we lower the output voltage of code 01 in DAC then, as Figure 5.2 shows, the intrinsic failure is prevented. Our cancellation scheme is based on this idea of distorting the DAC output to better fit the ADC transfer function.



DAC input code	ADC output code
000	00
001	00
010	01
011	10
100	10
101	10
110	11
111	11

Figure 5.3: Calibrate the nonlinear 2-bit ADC by a 3-bit DAC.

5.1.2 ADC Nonlinearity Calibration Method

To implement this idea, we integrate DAC with higher resolution than ADC in the test channel. In this way, we can adjust the output of DAC more precisely. Figure 5.3 illustrates how this modification works. The extra resolution in DAC enable it to generate much finer output levels. If we pick the DAC output levels which are marked by red, the ADC can correctly decode all the codes sent by the DAC. Intrinsic failure in the original case is thereby avoided. We did mathematical simulation based on the previous model to show the improvement of a 5-bit DAC and 4-bit ADC over 4-bit converter pair.

Figure 5.4 shows that the intrinsic failure rate is greatly reduced. Moreover, this calibration scheme can maximize the noise margin. Although this scheme adds to fixed cost of ATE test channels, in long turn, it benefits more by improving test application reliability. However, there are some kinds of errors in ADC which cannot be calibrated by this scheme, such as missing code and non-monotonicity. We mark such defective ADCs as MVL-incompatible. MVL-incompatible DUTs may function well but cannot receive correct test patterns from an MVL channel, for which they might not pass the test, resulting in yield loss. Our calibration scheme should be able

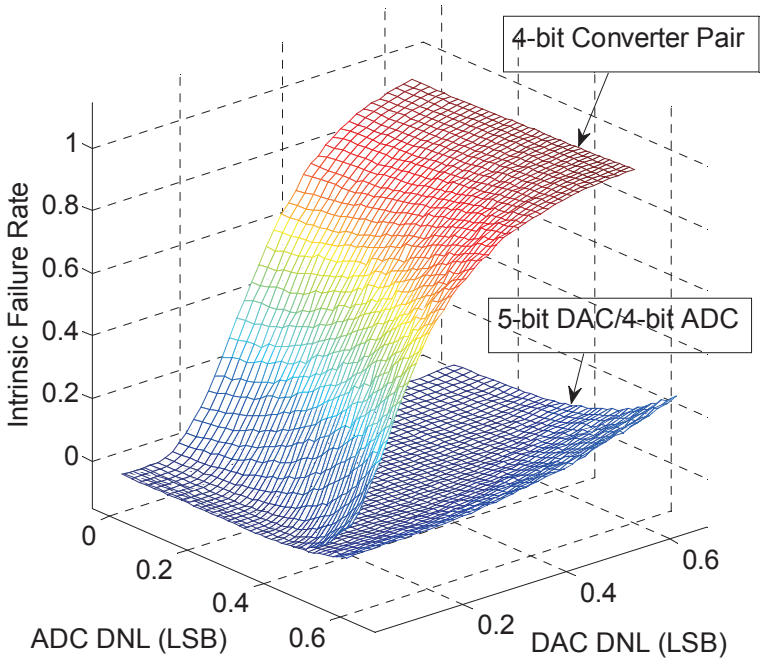


Figure 5.4: Intrinsic failure rate vs. ADC/DAC nonlinearities with/without extra resolution.

to identify the MVL-incompatible DUTs. Then we can have other method to apply tests to them, eliminating yield loss by malfunctioning ADC.

The calibration procedure has three steps, referred to as DAC input sweeping, feedback collection and DAC code redistribution or reporting calibration error. Using the example of Figure 5.3, the whole process is explained below.

- DAC input sweeping: For each test channel and DUT pair, a ramping-up pattern set is fed into the DAC to generate MVL stimulus. In this example, '000', '001', '010'..., '110' and '111' become the DAC input for each clock cycle. At the same time, the generated MVL signal is sent to the DUT.

- Feedback collection: The ADC in the DUT will capture the MVL ramping-up signal and decode it back into digital pattern each cycle. These decoded patterns will be sent back to the ATE through some receiving test channels.

- DAC code redistribution or MVL-incompatibility report: The decoded patterns are received by the ATE as normal testing responses. Then the ATE will select some particular codes as the calibrated codes for corresponding MVL channels. The selected codes should be the median of those have the same decoded pattern. For example, '011', '100' and '101' are all decoded as '10' in the instance. Then we pick '100' as the DAC input for code '10' as it's the middle one among three input codes. By this way, intrinsic failures are avoided and noise margins are maximized. When calibration cannot be done, this step will not redistribute DAC codes but report an error signal.

5.2 Error Control for Noise-Induced Errors

After taking care of the intrinsic failure, the problem of noise-induced error should be solved. Assume we have a DAC/ADC pair with certain level of *SER*. We can estimate the probability of the whole test being applied correctly as: $(1 - SER)^{\frac{T}{R}}$, in which T is the test data volume and R is ADC resolution. We name this probability as correct application rate (CAR). When the test data is incorrectly decoded, the test result is not credible, which may damage yield and increase test escape. So here we propose an error control scheme to detect incorrect test application and resend test data.

5.2.1 Examining Decoded Test Data and Retest

The basic idea of our error control scheme is to treat the MVL decoder as another DUT, whose output needs to be examined. In testing, there are two ways to examine test response: direct comparison of test response and signature based analysis. Direct comparison requires some test pins to continuously send back test responses, which is undesired in RPCT. However, the signature based analysis only needs to send back dozens of bits of test signature at end of the test. Therefore our error detection is

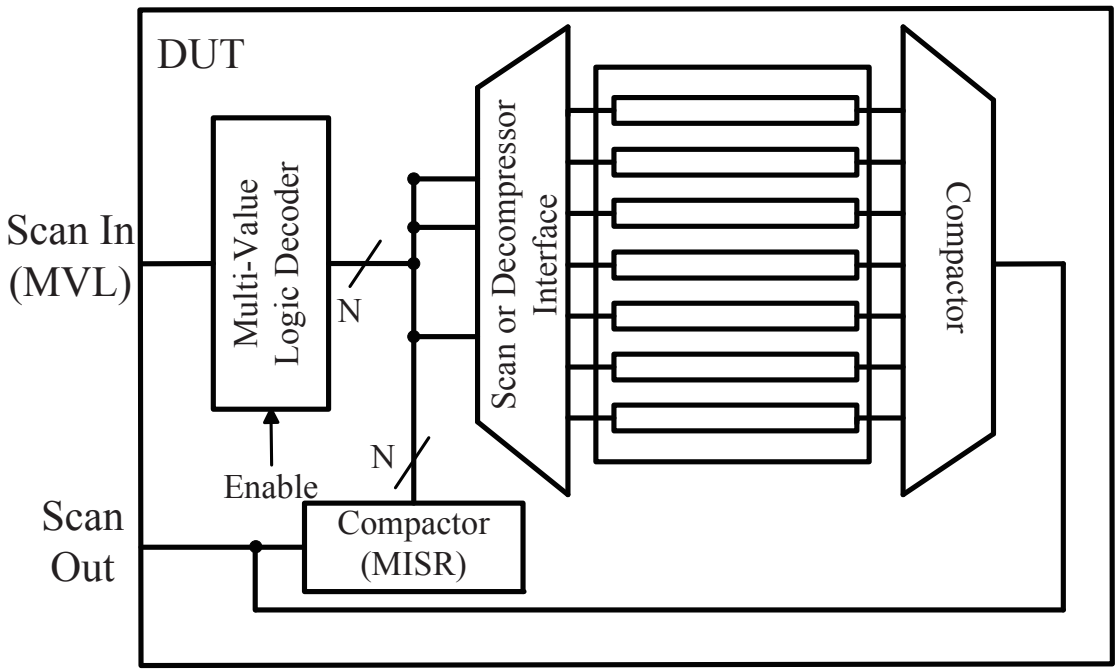


Figure 5.5: Test data application interface of DUT with MVL decoder and error control circuit.

based on signature analysis. Figure 5.5 gives a schematic view. An MISR is added and connected to the output of the ADC. So during the test application, the decoded test data are being compacted and sent to the scan or decompressor interface. When the test application is finished, the content in the newly added MISR becomes the signature for decoded test patterns, and we call it applied test signature (ATS). By examining this signature at the end of test, we will know if the DUT received correct test data. If the ATS matches the desired one, the test result of the DUT is verified. Then we can determine whether it passes the test. However, if the ATS does not match the desired signature (assume signature aliasing is ignored), this DUT must have received incorrect test patterns and the test result becomes unreliable. Then we require it to be tested again.

5.2.2 Mathematical Analysis of Reliability Improvement

Let CAR be the original correct application rate and N the number of retests. The correct application rate with retests can be calculated as:

$$CAR_{ec} = 1 - (1 - CAR)^{N+1} \tag{5.1}$$

The incorrect test application rate is suppressed by the power of retest times. We can equivalently consider the system SER to be lowered with retesting. The equivalent symbol error rate (SER') with retest time N is calculated as:

$$SER' = 1 - \{1 - [1 - (1 - SER)^{\frac{T}{R}}]^{N+1}\}^{\frac{R}{T}}, \tag{5.2}$$

in which T is the total test volume and R is the ADC resolution. We define the normalized SER improvement (I) as:

$$I = \log_{10} \frac{SER}{SER'} \tag{5.3}$$

We calculated I in different cases to show how it changes with N, SER and $\frac{T}{R}$. From the result shown in Figure 5.6, we can see that I increases with lower SER and higher N. Moreover, smaller $\frac{T}{R}$ ratio leads to more substantial improvement. In practical use, a maximum retest time should be set to avoid testing one device for too much time. If a DUT cannot pass the ATS examination with maximum times of retest, it will be categorized as MVL-incompatible. In conclusion, our error control scheme can effectively improve the reliability of MVL test application, which can be reflected by the increased equivalent SER. However, if the original SER is too high, we cannot gain a satisfactory improvement. What's more, factors like test data volume, ADC resolution and maximum retest time should be taken into consideration for particular testing task.

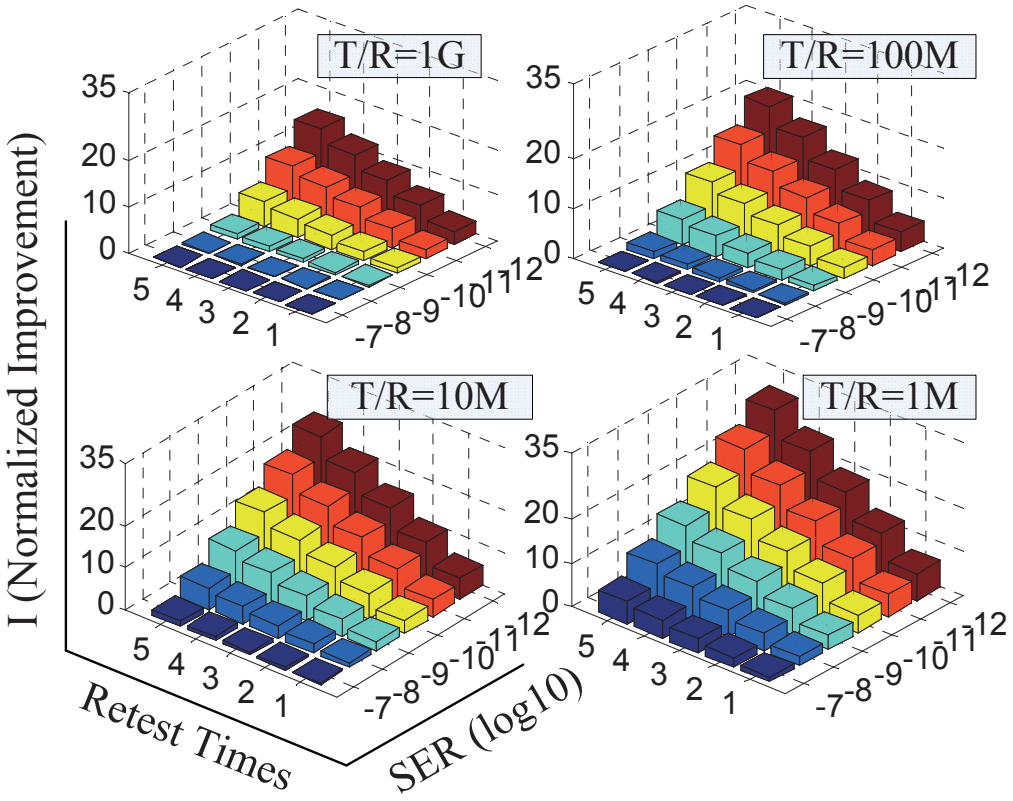


Figure 5.6: Normalized SER improvement vs. retest repetitions and SER with different test volumes.

5.3 MVL-Bypass Mode for MVL-Incompatible DUTs

In both calibration and test application procedures, some devices will be found as MVL-incompatible, which cannot get correct test stimulus through the MVL test channel. Because their test results are not credible, they are like untested devices. Passing them to a customer may result in defect escapes, and discarding them may result in yield loss. To have control on these devices, another test mode must be provided. We developed a so-called MVL-bypass mode for these devices. In this mode, the DUT receives test data in binary format, which skips the MVL decoder and directly feeds to the deserializer. However, due to the data rate reduction from switching back to binary data format, the test time will be longer, which has the same test time with traditional RPCT scheme. When MVL-bypass mode is activated, these devices can be tested by regular ATE with binary test channels.

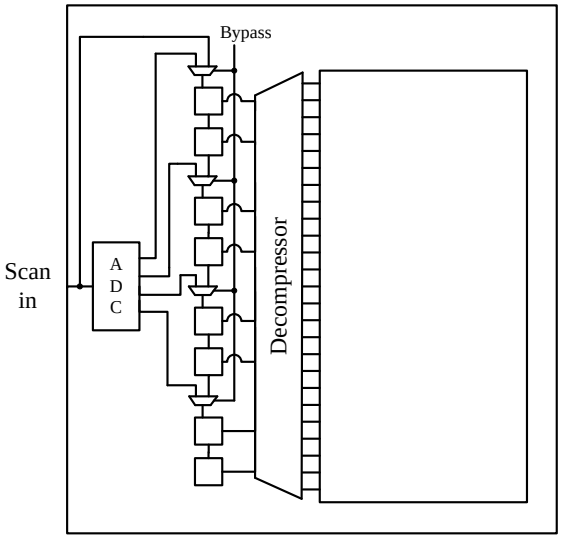


Figure 5.7: Simplified MVL-bypass mode implementation in DUT.

An illustration is shown in Figure 5.7. This diagram shows a DUT with a 4-bit ADC and an 8-bit decompressor interface. The 'Bypass' control signal will determine whether the DUT is in MVL or MVL-bypass mode. In MVL mode, the eight registers form four 2-bit long shift registers and supply inputs for the decompressor every two clock cycles. When it's in MVL-bypass mode, the ADC is bypassed by a direct connection from the test channel to the first register. This time, eight registers form only one shift registers to supply an input vector every eight clock cycles.

Chapter 6

MVL Test Hardware and Test Flow

6.1 Hardware Modifications on ATE

Our scheme requires test channels to send MVL format data and conduct ADC nonlinearity calibration, which cannot be achieved with the traditional ATE hardware. Therefore, modifications on ATE is needed.

6.1.1 Requirements on MVL-Compatible ATE

The ATE system needs some hardware modifications to generate MVL signals, which is implemented by integrating a DAC with each MVL channel as the MVL generator. To enable the ability of ADC nonlinearity calibration, calibration supporting circuitry should be integrated along with MVL channels.

The calibration support circuitry should be able to conduct all three calibration steps, which have been discussed in the previous chapter.

- Calibration step one: First of all, the support circuitry should provide a ramp-up pattern set to the DAC for ramp-up MVL signal generation, which is implemented as an up-count counter with the width the same as the DAC resolution. Notice that the ramp-up patterns cannot be provided from the test vectors because the MVL signal resolution is lower than the DAC resolution, for which the test vector cannot fill all the DAC inputs.

- Calibration step two: Besides, The support circuitry should also have the ability to process decoded patterns from receiving test channels and accordingly calculate the calibrated DAC input code for corresponding test vector.

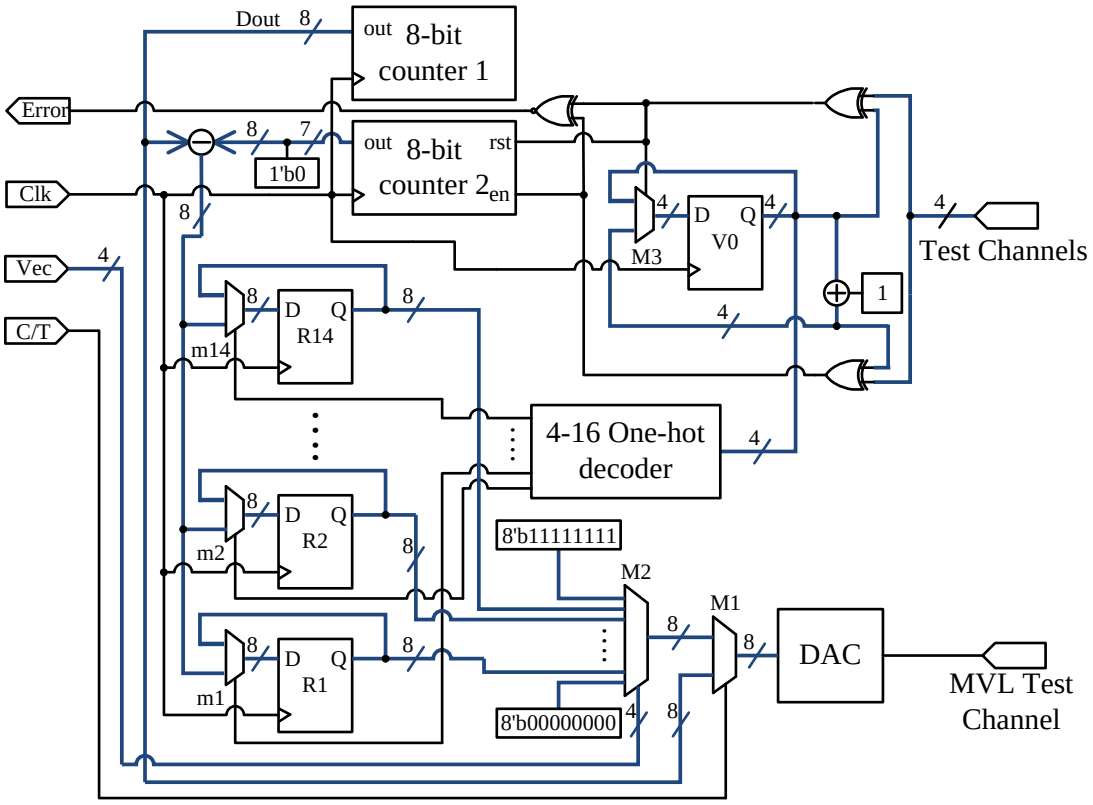


Figure 6.1: Modifications on ATE to support MVL signal and nonlinearity calibration.

- Calibration step three: At last, the support circuitry should have storage elements to store the calibrated DAC input codes. Therefore, these codes can be applied to the DAC to generate calibrated MVL signals during test. An error detection circuit should also be added to report the MVL-incompatible devices.

Because each MVL channel only interfaces to its own ADC, and each ADC has distinctive performance, every MVL channel should have a set of DAC and calibration supporting circuitry integrated.

6.1.2 Detailed MVL Channel Design

A detailed diagram of modifications of the ATE is shown in Figure 6.1. This implementation is based on an 8-bit DAC to generate a 4-bit MVL signal. Extra

4-bit resolution is used for ADC nonlinearity calibration. Next, I will introduce how this hardware design works to implement the calibration mode and test mode.

- Calibration mode:

  In calibration mode, control signal 'C/T' selects the lower channel of the 8-bit 2-to-1 multiplexer M1. The test vector 'Vec' will have no effect, since it is blocked by M1.

  - Calibration step one:

    The 8-bit counter 1 generates the ramp-up patterns and they are fed to the DAC through M1, so that a ramp-up MVL signal will appear at the MVL test channel.

  - Calibration step two:

    The 4-bit register bank V0 holds the value of the currently under calibration test vector. Initially, it's reset to all-0s. Every clock cycle, the decoded patterns are received by four binary channels, which are XORed with 'V0' and 'V0+1' separately. This part is to judge how received patterns change. If the received pattern equal 'V0', the XOR result will make V0 keep its current value for the next cycle and enable 8-bit counter 2 to up-count by 1. If the received pattern equals 'V0+1', the XOR result will make V0 update its value to 'V0+1' and reset 8-bit counter 2 to all-0s. Therefore, the 8-bit counter 2 can record how many cycles it takes for received patterns being increasing by 1, which is used to calculate the calibrated DAC input code. As we know, the calibrated DAC input code is the middle one of those applied DAC input patterns whose received patterns are identical. So this value can be calculated by subtracting half the 8-bit counter 2 value from the current applied DAC input pattern (value of 8-bit counter 1).

The halving of the value in 8-bit counter 2 is achieved by cutting off the LSB of the 8-bit counter and concatenate one '0' on its left.

- Calibration step three:

  According to the calibration algorithm, there should be 16 calibrated DAC input codes generated for 16 codes in 4-bit vector. To reduce the complexity, we don't assign calibrated values for '0000' and '1111' codes. Instead, we directly assign the minimum ('00000000') and maximum ('11111111') DAC input codes for them. Then we only need to store 14 calibrated codes in this case. As the calibrated value is generated by the subtractor, its output is fed to 14 8-bit register banks (R1-R14) to be stored. 'V0' is fed to a 4-to-16 one-hot decoder to generate control signal to select which register bank should be loaded with the calibrated DAC input code. For MVL-incompatible reporting, the comparison result between the decoded pattern and 'V0' is xnored with the comparison result between the decoded pattern and 'V0+1', generating the output 'Error' to report any MVL-incompatible devices.

- Test mode:

  In test mode, the upper channel of multiplexer M1 is active. Any 4-bit test data are applied as 'Vec', as all the calibrated DAC input codes have been stored in register banks. 'Vec' will control 8-bit 16-to-1 multiplexer M2 to send calibrated codes to the DAC, generating calibrated MVL signals on the MVL test channel.

6.2 Gate Level Implementation of ATE Modification

6.2.1 Hardware Implementation and Simulation

The above RTL design of the hardware is described in Verilog language, which is later synthesized to a gate level circuit. The synthesis library used is from ASIC

design kit (ADK) for TSMC250 technology. We supply the received patterns as the inputs given in a test bench. By modifying the received patterns, we can imitate different ADC responses to verify the functionality.

First of all, we supply the received patterns in a way in which regular ADCs would respond (not MVL-incompatible and have nonuniform response due to nonlinearity). The simulation result of this case is shown in Figure 6.2. In this waveform, the 'start' signal is used for reset register bank V0, which is not shown in hardware diagram. Signal 'Dout' is the ramp-up patterns generated by 8-bit counter 1. 'Vre' is the decoded patterns from the ADC. We make the duration of each code to not be the same to imitate nonlinear response. The 'error' signal stays low during the calibration, indicating the calibration is successfully done. At the end of the calibration, the calibrated codes are stored in R1-R14. In this case, the DAC has the calibrated input '00010101' to generate the MVL signal for code '0001', '00100111' for code '0010', '00111001' for code '0011', '01001010' for code '0100', '01011101' for code '0101', '01101100' for code '0110', '01111000' for code '0111', '10000100' for code '1000', '10010100' for code '1001', '10100101' for code '1010', '10110101' for code '1011', '11000110' for code '1100', '11010100' for code '1101' and '11100011' for code '1110'. For code '0000' and '1111', the calibrated inputs are preset to '00000000' and '11111111'.

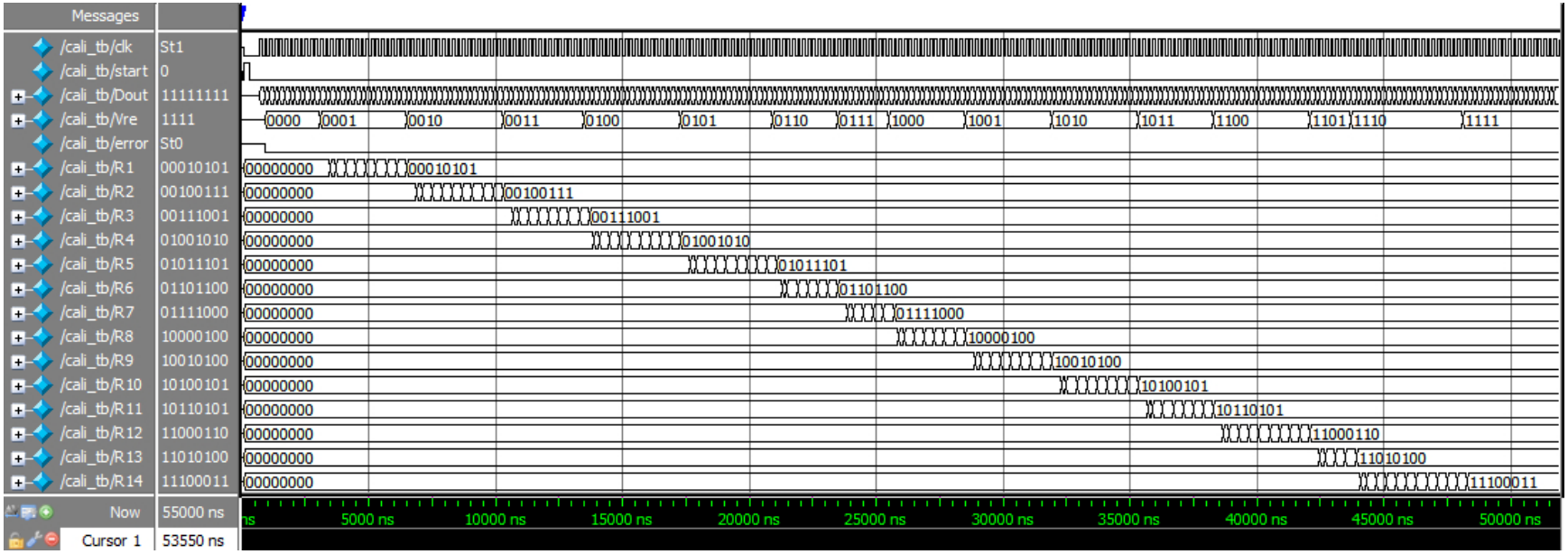


Figure 6.2: Simulation result of calibration behavior with gate-level implementation.

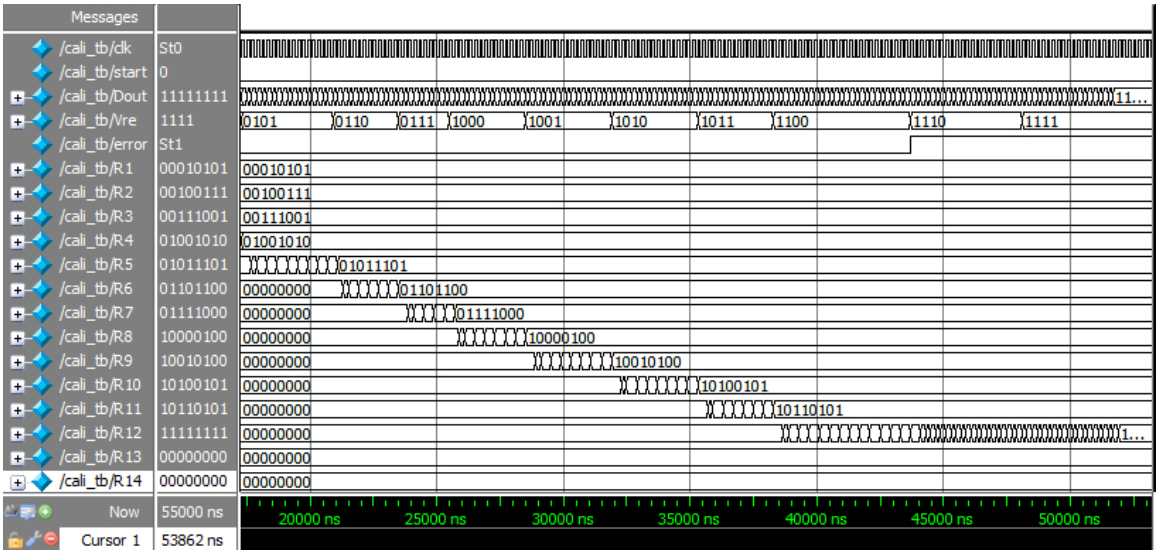


Figure 6.3: Simulation result of gate-level implementation detecting missing code.

We also did simulations to show how the calibration hardware detect MVL-incompatible devices. We use the same setup as the above simulation, except for some changes of the received patterns. We generate sample cases when missing code and non-monotonicity happen. First, the receive patterns are modified to have code '1101' missing. Figure 6.3 shows the simulation result. After receiving code '1100', the 'error' signal goes to high as it finds code '1101' is missing.

Next, we modify the received patterns to have the non-monotonicity property, making 'Vre' go back to '1011' after '1100'. In Figure 6.4, 'error' goes to high as it finds the non-monotonic code '1011'.

6.2.2 Extra Test Time and Area Overhead

The nonlinearity calibration procedure costs extra test time, whose number of clock cycles is equal to the length of the ramp-up pattern set. For an N-bit DAC, it takes 2^N clock cycles to apply all ramp-up patterns. In the above simulation, which uses an 8-bit DAC, it takes 256 clock cycles for all the calibrated DAC codes to be stored in register banks. Compared with the time to apply a large volume test data, it's only a very small portion.

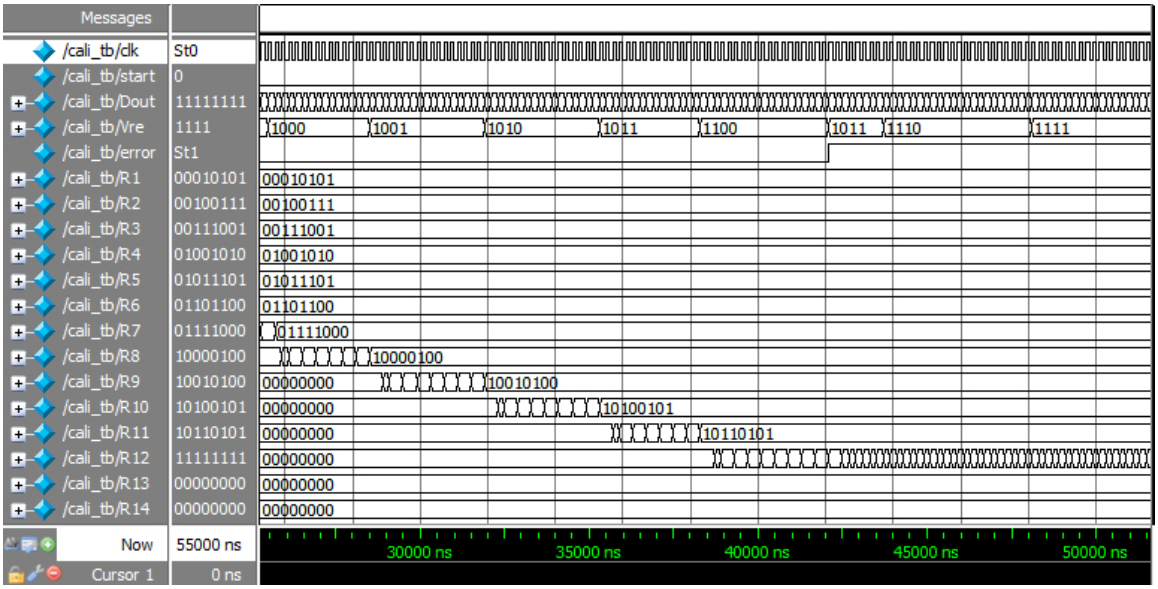


Figure 6.4: Simulation result of gate-level implementation detecting non-monotonicity.

This hardware implementation takes extra area overhead too. Except for integrating a DAC in each MVL channel, the calibration circuitry has a certain area overhead. The demonstrated calibration system takes 1028 unit gates. This number can be reduced to 723 if we replace all the register banks by random-access memory (RAM) cells.

6.3 Hardware Modification on DUT

In addition to the ATE, modifications are needed on the DUT to support MVL test application. First of all, an ADC should be built on chip, working as the MVL decoder. Then we integrate an MISR to be the ATS compactor for decoded test data. The DUT should also be modified to support testing with MVL channel and testing in MVL-incompatible mode. We use a single core design with decompressor interface to show the hardware modifications. The complete design is shown in Figure 6.5.

In the figure, we assume the MVL channel has 4-bit resolution and the decompressor has 8 inputs. An RPCT interface is connected to the decompressor, which

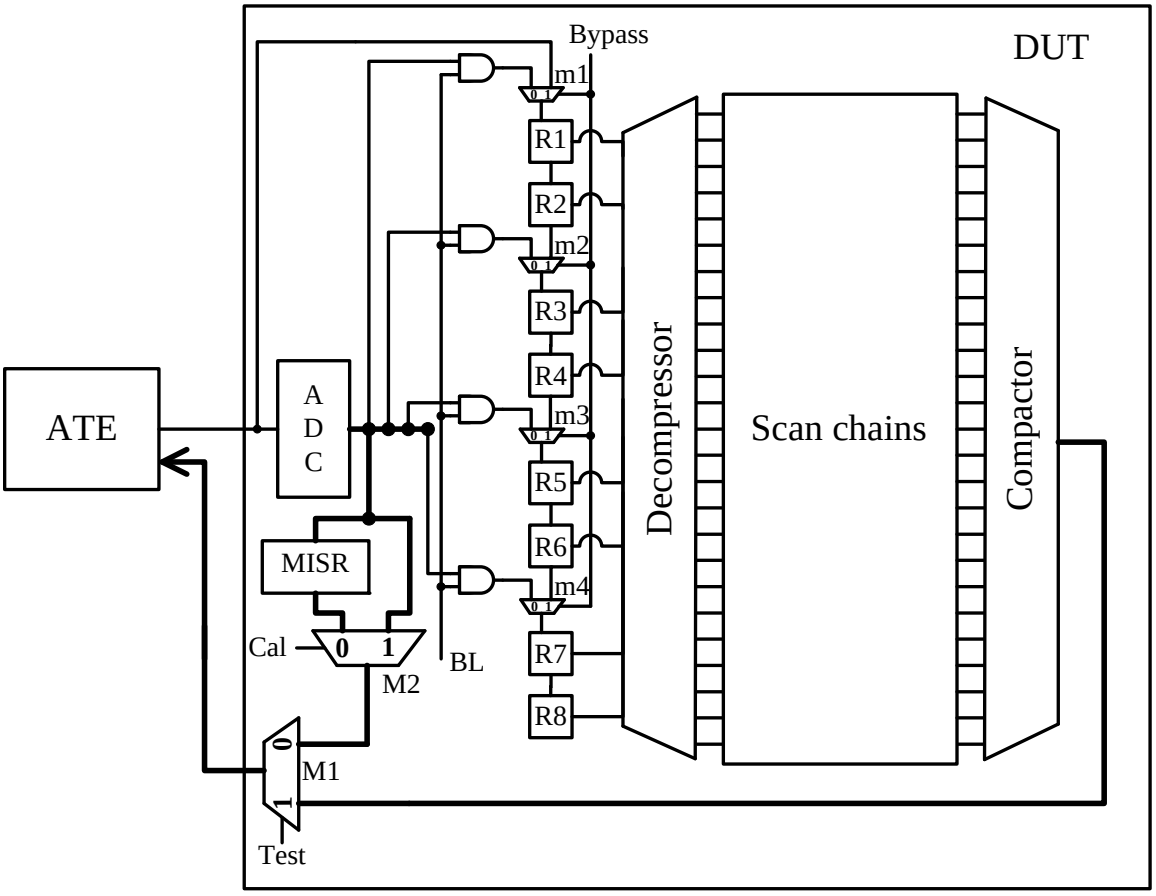


Figure 6.5: Detailed hardware modifications on DUT.

is made of 8 registers. To decode MVL signals, a 4-bit ADC is connected to the MVL channel. The outputs of the ADC are fed to the RPCT interface through some control logic. It's also connected to an MISR for ATS generation and bypassed to output for ADC nonlinearity calibration. To implement different functions, we have four control signals to configure the circuit. The 'Test' signal is for selecting between calibration mode and testing mode. The signal 'BL' is for blocking the ADC outputs from effecting logic under test during the calibration process. The 'Cal' signal is for selecting between reading ATS from the MISR or sending decoded patterns back for calibration. The last control signal 'Bypass' is for selecting between MVL signal testing or MVL-incompatible testing. Table 6.1 explains detailed configurations for different modes.

Table 6.1: Different configurations with control signals.

Configuration	Test	BL	Cal	Bypass
ADC Nonlinearity Calibration	0	0	1	0
MVL Test	1	1	X	0
ATS Capture	0	0	0	X
MVL-incompatible Test	1	0	X	1

- Configuration of ADC Nonlinearity Calibration:

  In this configuration, the control signals 'Test', 'BL', 'Cal' and 'Bypass' are '0010'. The blocking AND gates are activated by assigning 'BL' to be '0', which stops the decoded patterns from passing to the RPCT interface. The purpose of ADC output blocking in this configuration is to prevent the decoded ramp-up patterns from effecting the state of the DUT and causing unnecessary power consumption. Here, 'Bypass' is set to '0' in order to prevent RPCT interface capturing some unwanted data directly from the MVL channel. 'Cal' and 'Test' are set as '10' to enable the path from the ADC output to output pins, by which the decoded patterns can be directly sent back to the ATE.

- Configuration of MVL Test:

  The control signals are '11X0' in this configuration. This configuration allows the device to be tested with MVL channels. Blocking AND gates are disabled by assigning 'BL' to '1', so that the decoded test patterns can be sent to the RPCT interface. Also, setting 'Bypass' to '0' reforms the RPCT interface into four 2-bit shift registers. Control signal 'Test' connects the output of multiplexer M1 to the DUT compactor, which lets the DUT transfer test responses back to the ATE. 'Cal' is a don't-care in this configuration, since the output of multiplexer M2 has been blocked by M1.

- Configuration of ATS Capture:

  The control signals are '000X' for ATS capture. 'BL' blocks the ADC output from going to the RPCT interface. 'Bypass' is a don't-care for test activity being stopped. 'Test' and 'Cal' are set as '00' to send the output of MISR to the output pins, by which the ATE can capture the ATS and determine whether the decoded test is correct.

- Configuration of MVL-incompatible Test:

  To conduct MVL-incompatible test, the control signals are set to '10X1'. In this configuration, 'BL' is '1' to reduce the power consumption of the nodes following ADC outputs. 'Bypass' is assigned to be '1' to directly receive test data from test channel by the RPCT interface. Moreover, the 8 registers forms a single chain of shift registers. The 'Test' signal is '1' to connect the output pins to the DUT compactor, which send out the test response. Similar to configuration of MVL test, 'Cal' is a don't-care bit because multiplexer M1 has already blocked the output of M2.

As a result, we can see that with these hardware modifications on the DUT, the DUT can support multiple modes to meet different requirements which are discussed in the previous chapter.

6.4 MVL Test Flow

In the last section of this chapter, I will introduce the complete test flow design, which includes all the mentioned test configurations and is supported by above hardware modifications on the ATE and DUT. The test flow diagram is shown in Figure 6.6. The Test flow consists of following blocks.

- MVL channel calibration: In the beginning of our test flow, the MVL channel calibration should be done. The calibration algorithm will distinguish the DUTs

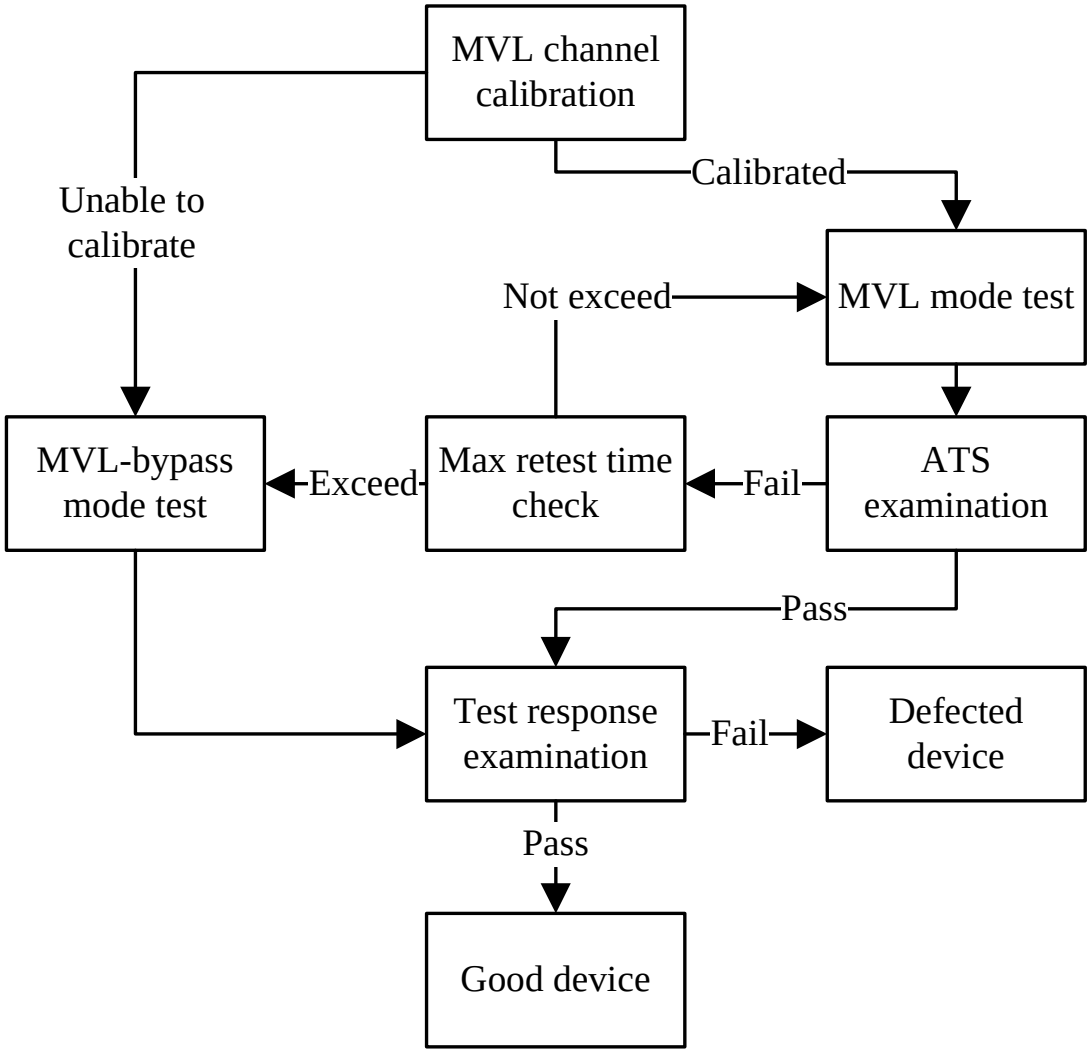


Figure 6.6: Complete test flow design with MVL test application.

which cannot be calibrated. These DUTs are categorized as MVL-incompatible, to be tested in MVL-bypass mode test. Those DUTs which can be calibrated are going to be tested in MVL mode test.

- MVL mode test: As the core of this work, the MVL mode test applies test data in MVL format. For each MVL channel and DUT pair, the MVL signal has been calibrated to achieve maximum noise margin. The test response is captured and analyzed as a regular test does. The DUTs to test in this step

have two sources: the DUTs which just get calibrated or those that are required to be retested.

- ATS examination: After MVL mode test, the first thing to be done is reading ATS and judging whether the DUT has the correct test data received. With this procedure, we avoid monitoring the decoded test data all the time, which causes extra requirements on test channel bandwidth. The test result is justified when it passes the ATS examination.

- Max retest time check: When the ATS examination turns out to be an unmatched result, there must be some errors during test data decoding. It's most probably caused by some unpredictable noise. Our scheme will require such DUTs to be retested. But before that, it needs to check how many times this DUT has been retested. Only if test time doesn't exceed some set maximum retest time, the retest will be conducted.

- MVL-bypass mode test: The DUTs to test in this step have two sources: the DUTs which are unable to calibrate in the MVL channel calibration step or those retested DUTs that have reached the maximum retest time. In this mode, the MVL decoder will be bypassed and binary test data are fed to run the test. The test result is credible for no MVL data being applied.

- Test response examination: The test responses of DUTs from MVL-bypass mode test and DUTs in MVL mode test with ATS examination passed are examined. This procedure decides whether the DUT is defected or good.

With this test flow, all DUTs are tried to be calibrated, all calibrated DUTs are tested with MVL channel and all MVL-incompatible DUTs are tested with binary test data. As a result, maximum effort is spent to have all DUTs tested by the MVL channel and no yield loss happens for MVL-incompatible DUTs.

Chapter 7

Reliability Measurement and Prototype MVL Test

To have an intuitive understanding of the reliability issue in MVL test data application, we measured the SER of a data converter pair, which is based on National Instruments Electronic Virtual Instrumentation Suite II+ (NI ELVIS II+) [12] prototype board. Here we picked an 8-bit DAC AD557 [2](DNL$\pm 0.5LSB$) and an 8-bit ADC AD7822 [3](DNL$\pm 0.75LSB$) from Analog Devices to work as the MVL generator and decoder. Like the way used in [95], the ELVIS system is controlled by Labview [10] program. The signal configuration and sent data are programmed and stored in a Labview program. Then any digital data can be sent to the ELVIS prototype board through a USB connection. In the meantime, the prototype board is able to capture any digital data and sent them back to PC for Labview program processing.

7.1 SER Measurement for General Data Converter Pair

In this experiment, we try to simulate:

- ATE sending MVL test signal:

  Pseudo-random test patterns are generated by a Labview program and sent to the MVL generator AD557 on the prototype board.

- DUT decoding MVL signal:

  The output of AD557 feeds into the MVL decoder AD7822, and the outputs of the AD7822 are wired back to the prototype board for response capture.

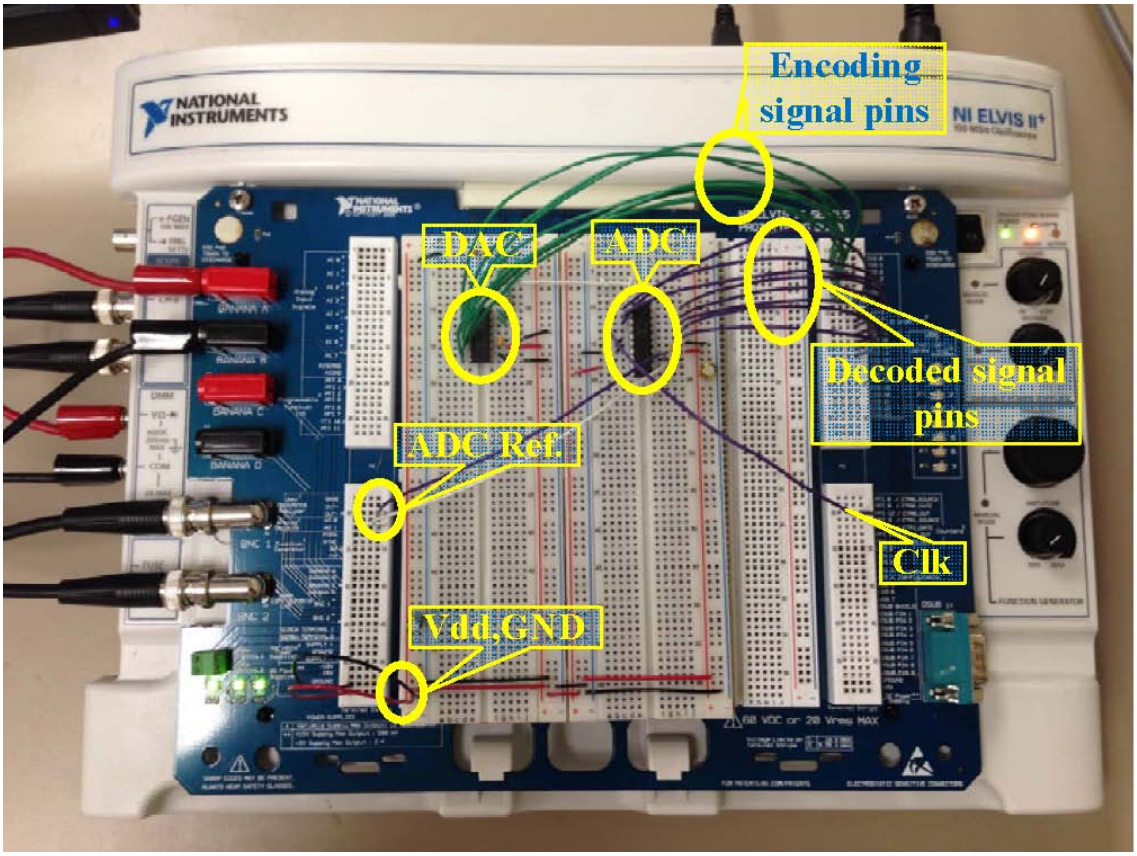


Figure 7.1: Test setup for SER measurement with NI ELVIS II+ and data converters.

- Captured data analysis:

  The captured decoded data are compared with sent patterns. Any mismatches are recorded to calculate SER.

The hardware setup for SER measurement is shown in Figure 7.1. In the figure, the encoding signal pins are connected to the DAC, which provide the pseudo-random patterns. The decoded signal pins are connected to the ADC, which captures the decoded patterns. There are other pins supplied by the ELVIS board such as ADC reference voltage, clock and power supplies. The data converter pair runs at

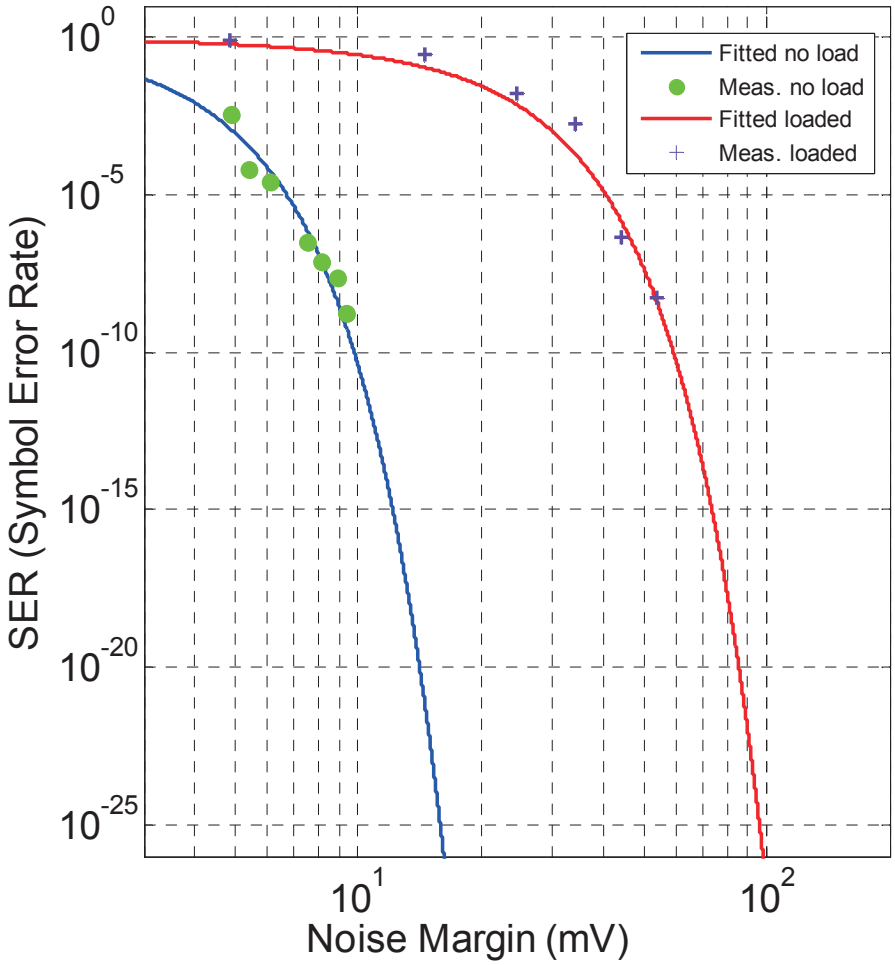


Figure 7.2: SER vs noise margin in prototype implementation.

0.4MHz because of hardware restrictions. The Labview program compares the original test data and the captured decoded data, calculating the decoding SER as $\frac{number\ of\ failed\ patterns}{number\ of\ sent\ patterns}$.

For the MVL signal, the noise margin can be calculated as $\frac{V_{full-scale}}{2^{bit+1}}$. In order to show the relationship between SER and noise margin, we use a voltage divider on the output of the MVL generator to change the full scale voltage of the MVL signal. By measuring SER with different noise margin, we get the relationship between SER and noise margin as shown in Figure 7.2.

There are two sets of measured data in this figure: 1) FPGA connected to the ADC, 2) no FPGA connected. The purpose of having an FPGA connected to the

ADC is to add digital noise in data conversions. In the figure, both sets of measured data are well fitted with the Gaussian Noise error function. We can also find that the requirement on noise margin is increased 5 to 6 times for FPGA-induced digital noise. However due to the slow conversion speed, it will take tremendously long time to measure low level SERs, which explains why all the measured SER are higher than 10^{-10}. Anyway, we can still predict SERs from the fitted curve. This data is also used in the following chapter as typical SER data for data converter pairs, for the purpose of performance estimation.

7.2 ADC Nonlinearity Calibration Method Verification

As introduced in Chapter 5 on assuring the test reliability, we proposed a calibration method to reduce the reliability damage from ADC nonlinearity. Here we conduct an experiment based on the prototype system to show the benefit of the nonlinearity calibration method.

In this experiment, we need to measure the SER in both calibrated and uncalibrated cases. Specifically, we need a pair of data converters with the same resolution as the reference group. Another DAC with higher resolution and the ADC in the reference group form the target group. Furthermore, the calibration method will be applied to the DAC in the target group. For better utilizing the available resources, we use the 8-bit data converter pair to create both reference group and target group. First, we emulate a nonlinear 5-bit DAC with the 8-bit DAC, in which 32 out of 256 output codes are non-uniformly picked to be the emulated DAC's output. The same method is applied to the 8-bit ADC for a nonlinear 5-bit ADC. Then we have the 5-bit converter pair as the reference group. The measured transfer function of the reference group is shown in Figure 7.3.

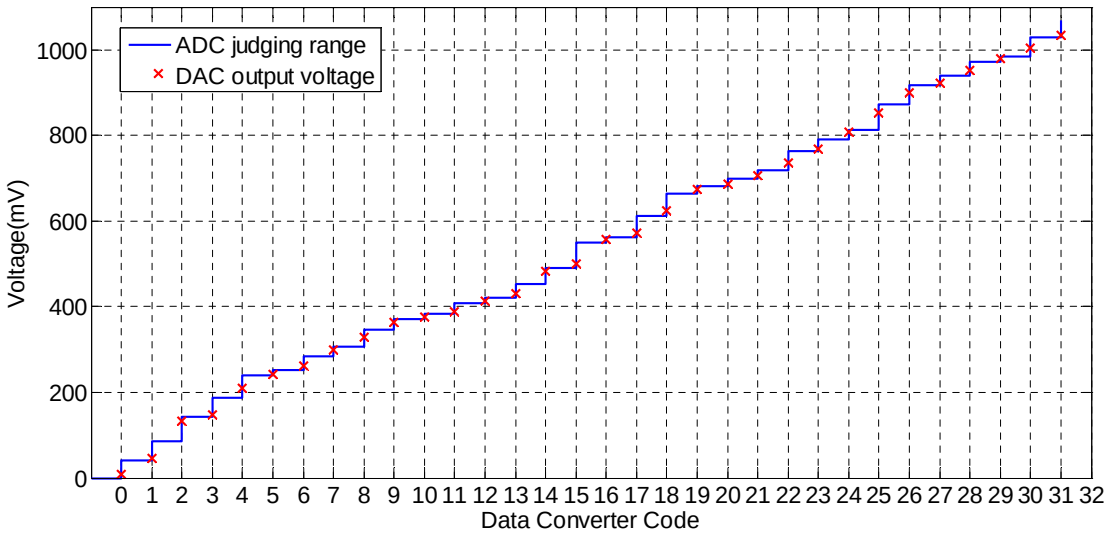


Figure 7.3: Transfer Function of the reference group.

Then we measure the SER of reference group with prototype system, in which case the SER is 1.88^{-2}. This SER is so high that it's impossible to send correct data with such converter pair.

Next, we use the calibration algorithm to repick 32 codes in the DAC to emulate the calibrated DAC. Now the target group consists of the new DAC and the previous ADC. We measure the transfer function of the target group as shown in Figure 7.4.

Compare Figure 7.3 and Figure 7.4; it's easy to observe that the DAC output voltage in the target group is much closer to the middle of the corresponding ADC judging range than what's in the reference group, indicating that the calibration method effectively increases the noise margin by only adjusting the DAC. Still, we measure the SER of the target group, which is 1.01^{-7}. It's five orders smaller than the result of the reference group, which demonstrates the calibration method does help in improving the reliability. It should be noted that if missing code or non-monotonicity happens, then it would be impossible to calibrate.

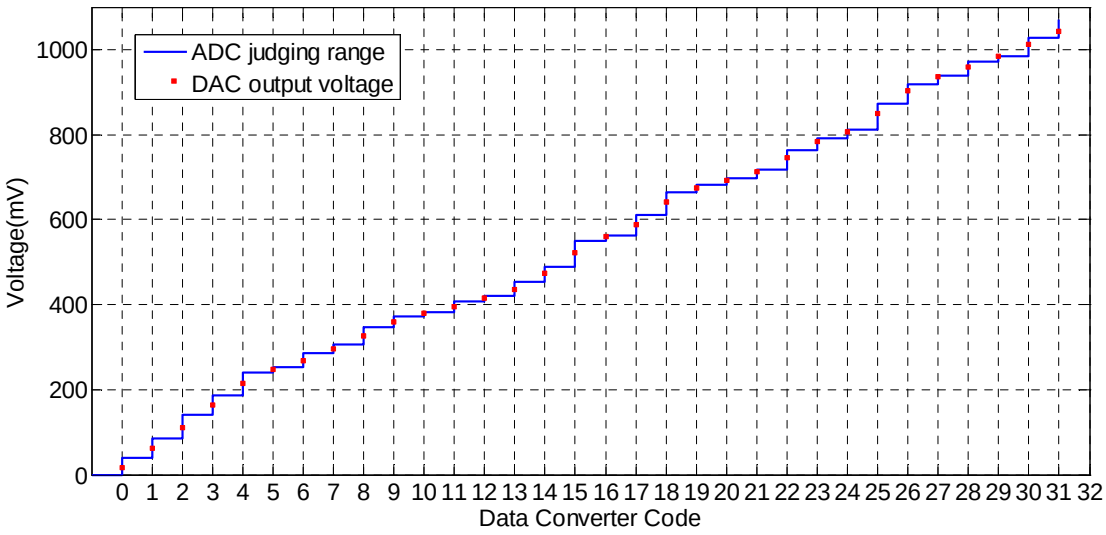


Figure 7.4: Transfer function of the target group.

7.3 Prototype Test With MVL Test Application

In this section, I'm going to demonstrate the concept of applying test data in MVL signal format by conducting scan-based test with the prototype system. Because the ELVIS system is able to send programmed or pre-stored digital data to the prototype board through the control of Labview program, it can play the role of a simplified digital tester. Feeding these digital data to the DAC, the DAC can generate a corresponding MVL signal. Together, the ELVIS system and the DAC work as a tester with an MVL test channel. To build a DUT receiving MVL test data, we first program a benchmark circuit into the Altera DE2 FPGA board as the logic under test. Then we place an ADC on the prototype board to receive the MVL signal for decoding. The output of the ADC is connected to the input of the FPGA board so that the ADC works as the MVL decoder. Figure 7.5 shows the experiment setup.

Compared with the experiment setup of SER measurement in Figure 7.1, the outputs of the ADC go to FPGA board instead of the receiving pins on the prototype board, providing the decoded digital patterns for the FPGA. The outputs of the

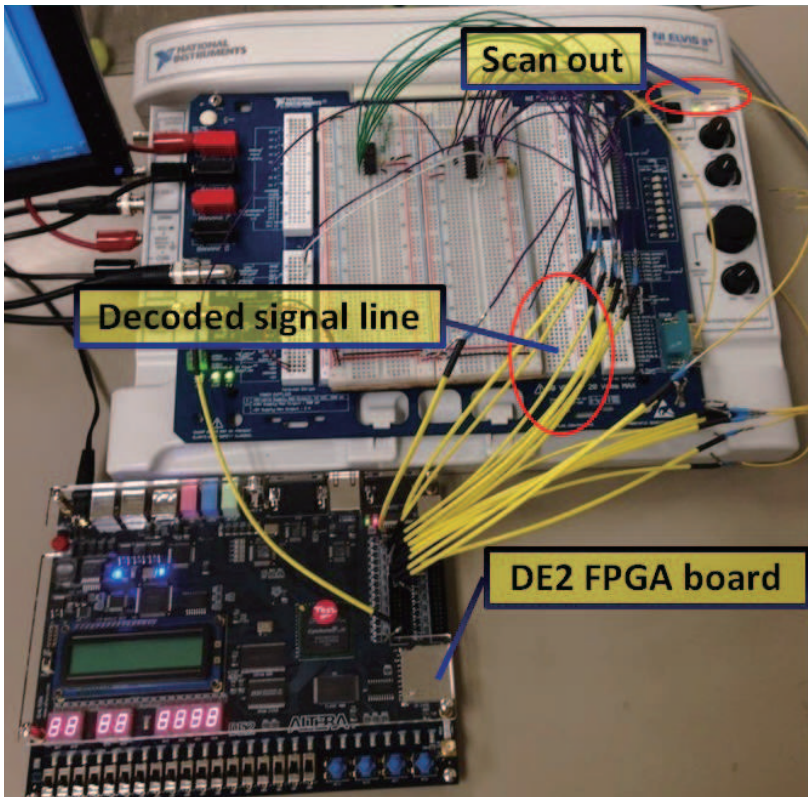


Figure 7.5: Test setup for scan test implementation with NI ELVIS II+, data converters and DE2 FPGA board.

FPGA board are connected to the receiving pins on the prototype board for response capture.

The core of our DUT is the ISCAS89 benchmark circuit s298 [8], which has been inserted with a single scan chain. It is programmed into the FPGA board. The test data for s298 contains 25 patterns. To apply each pattern, 14 vectors are sent to fill the scan chain. In total, 350 vectors are needed to complete the test, which are stored in Labview program. The benchmark circuit s298 has seven inputs: $/G0$, $/G1$, $/G2$, $/scan\_in1$, $/scan\_en$, $/blif\_clk\_net$ and $/blif\_reset\_net$. Five of them ($/G0$, $/G1$, $/G2$, $/scan\_in1$ and $/scan\_en$) are converted to one MVL signal. The other two inputs remains binary because the $/blif\_clk\_net$ is the clock signal which has two values in one period; and the $/blif\_reset\_net$ stays 0 during the whole test, so we just ground it. In this configuration, the 8-bit data converter pair is used to transfer 5-bit MVL

signal with approximately 8 times bigger noise margin, which increases the reliability. As a result, there are only two pins to receive test data for the emulated DUT, which are the MVL signal and $/blif\_clk\_net$. On the other hand, there are six outputs ($/G117$, $/G132$, $/G66$, $/G118$, $/G133$ and $/G67$) for s298. The output pins are all binary so that the emulated DUT has a total of eight pins connected to the prototype board.

The reference experiment was also done without MVL test data application. In this scenario, all six input pins are in binary format, which are provided by the prototype board directly. This time, there are 12 pins needed to be connected for the emulated DUT. The received test results are the same in both cases, which demonstrates the concept that we can apply the MVL format of data to conduct tests. Moreover, this comparison shows that testing with an MVL test channel uses fewer pins than the traditional way of test (in this case it is eight pins over 12 pins).

Chapter 8

Benefit Analysis of MVL Test Application

As discussed in Chapter 2, the use of MVL increases the amount of information contained in each symbol, so that the data rate of MVL communication can be much higher with the same symbol rate, which is determined by the clock frequency of the channel.

8.0.1 Ideal Data Rate Improvement

The data rate in traditional binary transmission equals the channel clock frequency. By converting the media signal into MVL format, the data rate of each channel is the channel clock frequency times the resolution of the MVL signal, in this case, the on-chip ADC resolution. We denote the data rate in binary communication as R_b, which can be calculated as:

$$R_b = f_{clk} \cdot \log_2 2 = f_{clk}, \tag{8.1}$$

where f_{clk} is the channel clock frequency and there are 2 levels in a binary symbol. Next, we calculate the data rate in MVL communication (R_{mvl}) as:

$$R_{mvl} = f_{clk} \cdot \log_2 N = f_{clk} \cdot R, \tag{8.2}$$

in which there are N levels in an MVL symbol and R refers to MVL signal resolution. If we assume the same channel clock frequency for both binary and MVL

communication, the data rate improvement (ϕ_i) in the ideal case can be shown as:

$$\phi_i = \frac{R_{mvl}}{R_b} = R, \tag{8.3}$$

in which the ideal case indicates no errors happening in both binary and MVL communication.

For the current communication technology, it is assumed that the binary data transmission is reliable in ordinary conditions. On the other hand, the MVL communication is not reliable, so that protection methods should be applied for it to increase the reliability, which has been well discussed in previous chapters. In this work, we use an error detection and retest scheme to deal with it. However, retest consumes more time to correctly send the same amount of data. MVL-incompatible devices need to be tested with binary signals after a maximum repetition of MVL mode tests, which completely loses the advantage over the traditional case. These situations increases the average time to test a DUT, which equivalently lowers the channel data rate. In the next section, analysis of data rate improvement with retest and MVL-incompatible devices considered is given to show the benefit more realistically.

8.0.2 Data Rate Improvement in a Real Condition

In this part of the analysis, we ignore MVL-incompatible devices which fail at the nonlinearity calibration stage, because such cases are very rare and the causes of them are usually unpredictable.

It's easy to know that each retest consumes a fixed length of time and the time to test MVL-incompatible devices is fixed as well. The key factors in determining real data rate improvement depend on:

- What's the chance for a retest to happen?
- How many times should retest be repeated?

- How many devices are MVL-incompatible after a maximum repetition of retest?

All the above factors are related to the reliability level of MVL communication. CAR, the correct application rate, is calculated by the following equation:

$$CAR = (1 - SER)^{\frac{T}{R}}, \tag{8.4}$$

where T is the volume of test data and R is the MVL signal resolution. The physical meaning of CAR is the chance to correctly sent T bits of data in R-bit MVL format with symbol error rate SER. Also, we define the time to send all data in MVL format as t_{mvl}, and t_b in binary format. We can easily get $t_{mvl} = R \cdot t_b$ with the same number of physical channels. First we calculate the chance of MVL-incompatible test ($P1$) as:

$$P1 = (1 - CAR)^{N+1}, \tag{8.5}$$

where N is the maximum retest time. This portion of devices need to be tested in MVL test mode for $N + 1$ times and MVL-incompatible mode for one time. So the time expectation for this portion ($T1$) is:

$$T1 = (1 - CAR)^{N+1}[(N + 1)t_{mvl} + t_b]. \tag{8.6}$$

For those devices which do not need to be tested in MVL-incompatible mode, the chance of conducting n times ($n \leq N + 1$) of MVL test (P_n) is:

$$P_n = CAR(1 - CAR)^{n-1}. \tag{8.7}$$

We know that the corresponding test application time for P_n is nt_{mvl} so that we can get the time expectation for a device to receive correct data within maximum

times of retest ($T2$) as:

$$T2 = t_{mvl} CAR \sum_{n=1}^{N+1} n(1 - CAR)^{n-1}. \tag{8.8}$$

Then we can get the time expectation to test a random device in real condition (T_{real}) as:

$$\begin{aligned} T_{real} &= T1 + T2 \\ &= (1 - CAR)^{N+1}[(N+1)t_{mvl} + t_b] + t_{mvl} CAR \sum_{n=1}^{N+1} n(1 - CAR)^{n-1}. \end{aligned} \tag{8.9}$$

Finally, we can have the data rate improvement in real conditions (ϕ_{real}) as,

$$\begin{aligned} \phi_{real} &= \frac{\frac{1}{T_{real}}}{\frac{1}{t_b}} \\ &= \frac{R}{(1 - CAR)^{N+1}(N + 1 + R) + CAR \sum_{n=1}^{N+1} n(1 - CAR)^{n-1}}. \end{aligned} \tag{8.10}$$

For a more intuitive illustration, the graph in Figure 8.1 gives a relationship between ϕ_{real} and CAR for a sample case where we assume that R is 5 and N is 4. We observe that ϕ_{real} equals R only when CAR is 1, which is the ideal situation. In the worst case when all devices need MVL-incompatible test ($CAR = 0$), ϕ_{real} is the lowest as 0.5, which is even worse than the traditional test method. Besides, it also shows that ϕ_{real} and CAR have a near linear function and the minimum CAR for the system to have benefit is around 0.2. Therefore, we definitely want CAR as high as possible to have better data rate improvement.

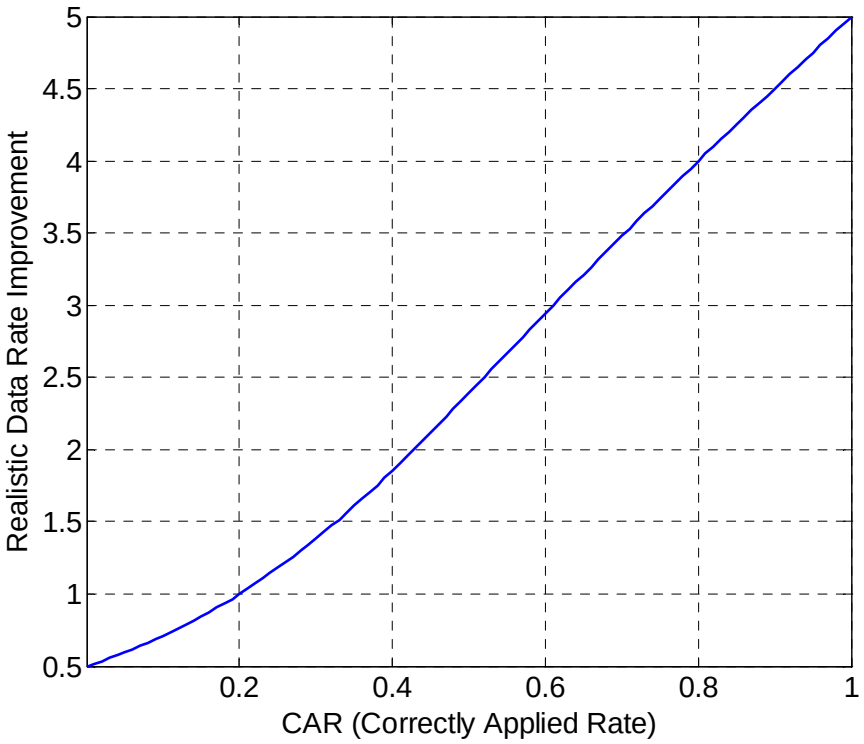


Figure 8.1: Realistic data rate improvement vs. CAR.

8.1 How to Configure MVL Channels

As discussed in the previous section, a higher value of CAR is crucial to achieving high MVL channel data rate. We also know that CAR is determined by system SER and the number of symbols to be sent ($\frac{T}{R}$) in Equation 8.4. Usually we can't change the test size T but we can choose ADC resolution R and have methods to adjust system SER. It should be noted that the choosing of ADC resolution is also an important way to adjust SER; meanwhile it affects the value of $\frac{T}{R}$, because the ADC resolution determines the noise margin. To simplify the problem, we assume a fixed noise level in MVL data communication. Therefore, we only have control of the MVL signal resolution and full-scale voltage. However, it's still a complicated problem with many variables interacting with each other and having a combined impact on CAR. Therefore, I will use an example analysis to show how we sort this problem out, finding the optimum variables combination for the best data rate improvement.

Table 8.1: SER for different test size (T) and resolution (R) to reach 90% CAR and corresponding ϕ_{real}.

T(bit) \ R	3	4	5
10^6	3.16×10^{-7}	4.21×10^{-7}	5.27×10^{-7}
10^8	3.16×10^{-9}	4.21×10^{-9}	5.27×10^{-9}
10^{10}	3.16×10^{-11}	4.21×10^{-11}	5.27×10^{-11}
ϕ_{real}	2.7	3.6	4.5

8.1.1 Configuring the MVL Channel

Here we provide an example case to analyze, which has already had some preset testing requirements for the test scheme:

- The maximum retest time (N) is 4.
- 99.999% of DUTs are required to receive correct test data in MVL test mode, which means less than 0.001% DUTs are MVL-incompatible.
- The test size (T) of possible tests are 10^6 bits, 10^8 bits and 10^{10} bits.
- ADCs with 3-bit, 4-bit or 5-bit resolution are available.
- The maximum full-scale voltage of MVL signal is 1.5V because of limits on power supply and ADC technology.

With the first two requirements, we can calculate the needed CAR as 90%. Next, we calculate the required SER and data rate improvement in real conditions (ϕ_{real}), corresponding to different test size (T) and resolution (R), which is shown in Table 8.1. From this table, we know the needed SER level for each configuration to meet the given requirements. To connect these SER values to the hardware requirements, we use the representative data in Figure 7.2. By looking up the noise margin of

Table 8.2: Minimum noise margin and full-scale voltage for each case in Table 8.1.

43.9mV / 702.4mV	**43.4mV / 1388.8mV**	$46.9mV/3001.6mV$
51.3mV / 820.8mV	$50.8mV/1625.6mV$	$50.5mV/3232mV$
57.8mV / 924.8mV	$57.4mV/1836.8mV$	$57.1mV/3654.4mV$

corresponding SER from the curve with FPGA load in the figure, we can derive Table 8.2 showing the required noise margin (left side value) in each case.

With the noise margin known, we calculate the full-scale voltage in each case, which is also shown in Table 8.2 (right side value). According to the last requirement of maximum full-scale voltage 1.5V, there are only four cases satisfying it in the table, which are marked in bold. As the result, the best achievable ϕ_{real} for test size 10^8 and 10^{10} bits is 2.7 with a 3-bit ADC, and the best achievable ϕ_{real} for test size 10^6 bits is 3.6 with a 4-bit ADC. The we can choose the ADC with most benefit (3-bit for test sizes 10^8 and 10^{10}; 4-bit for test size 10^6).

The analysis of this example gives a way to find the MVL configuration for the best achievable data rate improvement.

8.2 Benefit of MVL Test Application for Devices with RPCT and Test Compression

Two relevant DFT techniques are:

- Test compression: Except for BIST-based test plans, test compression is almost a required test-related technology. The benefit of test compression is so significant that the test volume and time can be reduced by up to 100x.

- Test pin reduction: On the other hand, for better supporting multi-site test, technologies for reducing test pins are also quite popular. There are some literatures talking about test pins reduction, such as much earlier work [99] to more

recent ones such as [62], [30], [76] and [27]. These works are named differently, such as RPCT, LPCT(Low-Pin-Count Test) and Pin-limited Test, etc. But they have the same basic idea to time-multiplex the test data through fewer test channels, which sends serialized test data from the ATE and deserializes them in the DUT. In [27], the author discussed how RPCT benefits test compression based design in detail. The major benefit comes from avoiding fault coverage loss, so that less top-off patterns are needed in bypass mode, resulting in total test volume and time reduction.

8.2.1 Integrating RPCT-Compatibility for Devices with Test Compression and RPCT

MVL test application greatly increases the test speed to test RPCT and test compression based devices, especially for multi-core SoCs. Here I propose the use of MVL test application with RPCT interface. Associated to the instance in Figure 1.7, the proposed structure is shown in Figure 8.2. The RPCT interface of an RPCT-only scheme concatenates decompressor inputs of four cores into one chain with 20 cells. Like in the previous analysis, the internal scan speed is just $\frac{f}{20}$. When the test channel is replaced by a 4-bit MVL channel and the MVL decoder is integrated in the DUT, the long chain is broken into four chains with five cells, and the output of MVL decoder feeds into each chain. As a result, the internal scan speed is now $\frac{f}{5}$, which is improved by 4x.

When test data are supplied with MVL test channel, the long RPCT interface can be divided into several shorter ones, therefore it requires fewer test clock cycles to match the bandwidths of test channel and RPCT interface, speeding up the internal scan speed.

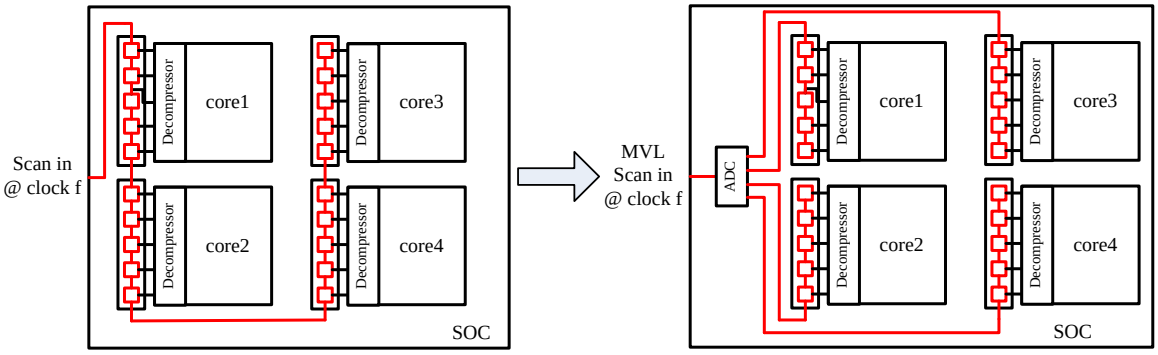


Figure 8.2: MVL test application for multi-core SoC with RPCT interface and decompressor.

8.2.2 Performance Comparison Between RPCT only and RPCT with MVL Test Applications

To intuitively show the benefit of the scheme of combining RPCT with MVL test application, we generated real tests for some DUTs and compared the test time reduction. The DUT we chose is ITC99 benchmark circuit [9] b19. We use Testkompress [11] to insert EDT logic for b19. The inserted blocks include a decompressor, an X-mask and a test response compactor. After EDT insertion, the DUT has 6.6K registers and 59K faults. Then we generate compressed test patterns by Testkompress and top-off test patterns in bypass mode with Fastscan. We list five possible configurations for comparison, which are shown in Table 8.3.

The first configuration is the original full scan design with one scan chain. The second one is EDT compression inserted. The third one is EDT compression inserted with RPCT interface. The fourth one is EDT compression inserted with MVL test application. The last case is EDT compression inserted with MVL test application and RPCT interface. All configurations assume to be tested with one physical channel and the MVL channel is in 4-bit resolution. The third column refers to the width of the EDT decompressor input. From the original fault coverage in Table 8.3, we can see that smaller width of EDT decompressor input, results in less fault coverage with the

Table 8.3: Test volume and time reduction in different scenarios.

Config.	Channel Type	Test Interf. BW	Orig. Test Vol.	Bypass Test Vol.	Orig. FC	FC w/ Bypass Test	Test Vol. Reduc.	Test Time Reduc.
Full scan	binary	1	4.4M	-	99.6%	-	-	-
Comp.	binary	1	333.4K	1.5M	93.5%	99.6%	2.33x	2.33x
RPCT+ comp. [27]	binary	4	666.4K	981.5K	95.6%	99.6%	2.67x	2.67x
	binary	8	710.8K	661.8K	95.7%	99.6%	3.21x	3.21x
	binary	12	750.6K	479.2K	95.7%	99.6%	3.58x	3.58x
	binary	16	707.6K	471.7K	95.7%	99.6%	3.73x	3.73x
MVL+comp.	MVL	4	666.4K	981.5K	95.6%	99.6%	2.67x	10.68x
MVL+ RPCT+ comp.*	MVL	8	710.8K	661.8K	95.7%	99.6%	3.21x	12.84x
	MVL	12	750.6K	479.2K	95.7%	99.6%	3.58x	14.32x
	MVL	16	707.6K	471.7K	95.7%	99.6%	3.73x	14.92x

compressed test patterns. This effect becomes insignificant when the decompressor width exceeds 8. From the data in the fifth column, we can see the major cause of test volume increasing is from the bypass patterns. Also, the column of "Test Vol. Reduc." tells that the reduction in test volume in [27] and this work are the same. However, according to the data in the last column, it shows that our scheme can conduct tests at four times the speed of the approach in [27].

Chapter 9

MVL Test Using RPCT Interface on ATE Platform

In the previous chapter, we showed a conceptional MVL test application implementation and analyzed the data rate benefits with practical considerations. In this chapter, to further demonstrate the idea of MVL signal based testing, we endow the real ATE system with the capability to generate MVL signals. Furthermore, we use this ATE system to test a MVL-compatible DUT with RPCT interface, showing the reduction in test time and channels from the combination of MVL test application and RPCT technology.

9.1 Implementing MVL Channel on Advantest T2000GS ATE System

Hardware Description

The tester we worked on is the T2000GS from Advantest. This tester is available at the department of Electrical and Computer Engineering in Auburn University.

Functionally, the T2000 system consists of 3 major parts: main unit cabinet (mainframe), test head and user interface (keyboard, mouse and monitor).

- Main unit cabinet:

  The main unit cabinet holds the system and site controllers, power supplies and interconnecting buses. The site controller send orders to control different modules in the test head.

- Test head:

  The test head has the ability to mount multiple test modules for different testing purposes, such as sending and receiving digital or analog signal, measuring

voltage and current and powering the DUTs, etc. These test modules are plug-in boards which makes it easy to configure the tester.

- User interface:

  The user interface consists of a keyboard and mouse set and a monitor, which enables the user to program the test activity with a PC-like interface.

Test Program Description

To configure and conduct test behavior, a test program is required, which is based on open architecture test programming language (OPTL) [14].

A complete test program includes a set of files:

- *.bdefs: bin definitions description.
- *.ctyp: custom type description.
- *.fh: description of pre-headers for custom function.
- *.lvl: voltage levels assignment.
- *.pat: provide the test patterns.
- *.ph: description of pre-headers for test classes.
- *.plist: assign the used .pat files in test flow.
- *.pxr: set timing maps, registers and resource selectors.
- *.scf: description of system controller flows.
- *.spec: describe specifications for test.
- *.tcg: describe test condition group.
- *.tim: describe timing behaviors.

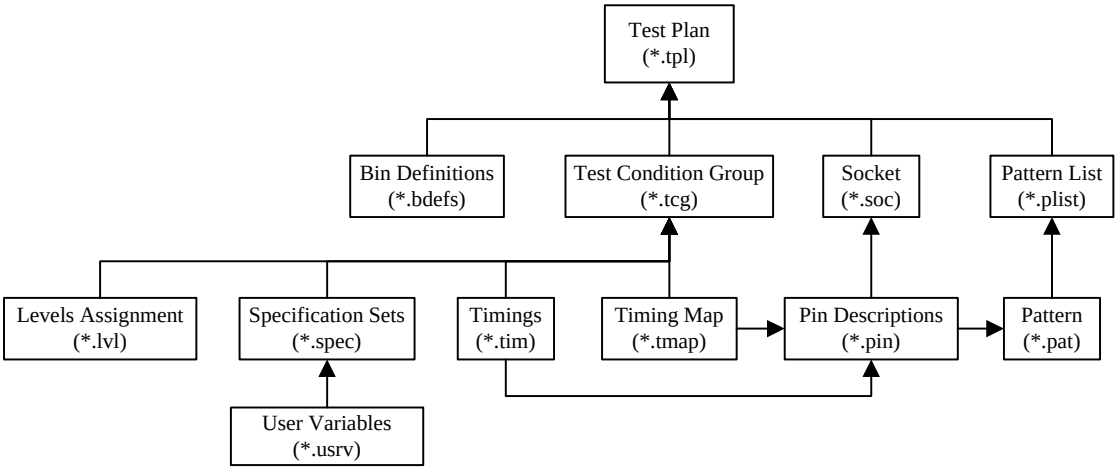


Figure 9.1: Relationship of used files in our test program.

- *.tmap: map timing description to test flow.
- *.usrv: define user variables.

A flow chart to show the relationship of the files used in our test program is given in Figure 9.1.

9.1.1 Building the MVL Channel

The test modules involved in regular digital test are 250MHz digital logic test module (250MHz DM) and 500mA device power supply (500mA DPS). The 250Mhz DM has 128 I/O channels, which are capable of driving and comparing signal pins. It can apply digital test data to the DUT and capture the test response from the DUT. The 500mA DPS supports the DUTs with 32 channels of power supply. The generated voltage range is from -2V to 12V and the resolution is 1mV. With this module, we can power the DUT according to its specifications. To build the MVL channel, we also used these two modules.

Notice that the 500mA DPS is capable of generating 32 different voltage levels, which can be used to provide voltage levels for MVL signals. 32 voltage levels can support MVL signals with 5-bit resolution. Because some DPS channels are needed

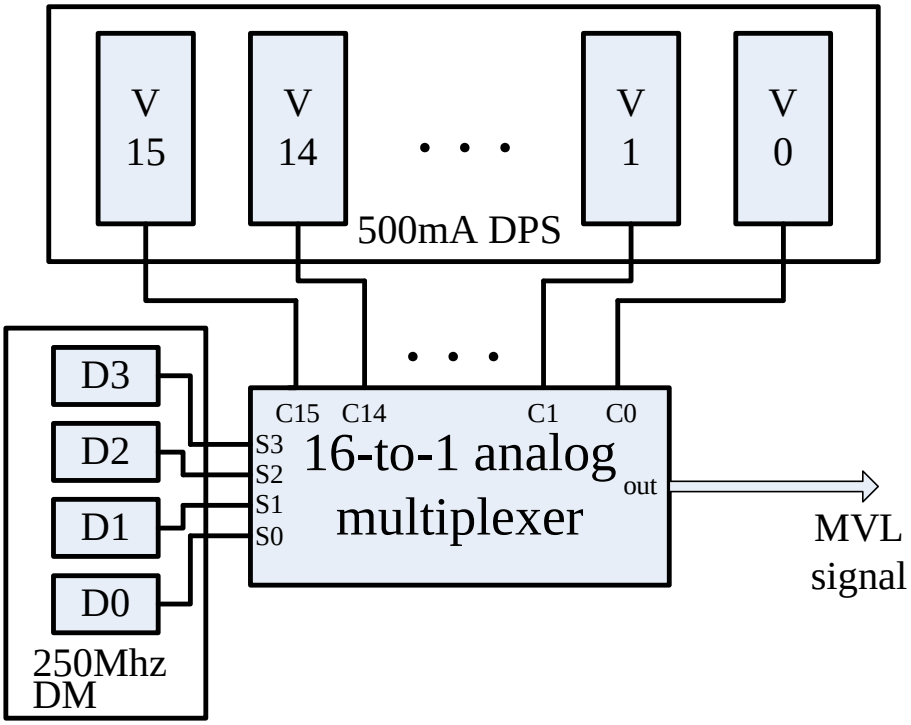


Figure 9.2: Generation of MVL signal with 500mA DPS and 250MHz DM.

to power up the DUT, we decided to use 16 500mA DPS channels to generate voltage levels for a 4-bit MVL signal and use the left channels for power supply. When the 500mA DPS channels are activated, the generated voltage levels are consistently presented on the terminal heads. In order to create an MVL signal, we need to multiplex these 16 voltage levels. Therefore an analog multiplexer is used to do the job. The analog multiplexer is 16-to-1 controlled by 4 selecting bits. In such a way, the information of the 4-bit MVL signal is actually conveyed by four selecting bits, which are driven by the I/O channels from 250Mhz DM. The MVL channel implementation diagram is shown in Figure 9.2.

For this MVL channel implementation, the test data is applied by four 250Mhz DM channels (D3-D0 in the diagram), and the voltage levels of the MVL signal are from the 500mA DPS (V15-V0 in the diagram). As a result, we built up a 4-bit MVL channel with the T2000GS ATE system by adding an additional analog multiplexer.

9.2 Building MVL-Compatible DUT with RPCT Interface

In the previous chapter, we verified the concept of MVL test application with the ELVIS prototype board based experiment. The DUT was the benchmark circuit s298 with a single scan chain. To justify our scheme, which combines RPCT with MVL test application, a new DUT should be built. The way to build the new DUT is similar to the simulated case in Section 7.3, but this time we need it to be implemented as a real device. We picked ISCAS89 benchmark circuit s38584 [8] as the logic under test. Twenty-eight scan chains are inserted by using Dftadvisor. EDT insertion is completed by Testkompress [11] with five EDT channels, meaning the input width of decompressor and output width of compactor is 5-bit. The s38584 with EDT inserted has total 23 inputs, including 15 primary inputs, five decompressor inputs and three control signals (*edt_clock*, $blif\_clk\_net$ and $blif\_reset\_net$). Because the DUT is synchronized by the three control signals, there are 20 inputs left to be sent in MVL format. As stated before, we have a 4-bit MVL channel available, therefore the 20 inputs should be driven by a 4-to-20 RPCT interface, to be loaded with the four outputs of MVL decoder. The structural view of the DUT is shown in Figure 9.3.

The size of the Testkompress generated test is 59.2Kb which consists of 12351 vectors. In the RPCT only scheme, the 4-bit ADC is removed. Instead, a 1-to-20 RPCT interface is used to receive test data through a binary channel. As a result, it takes 20 cycles for the RPCT interface to receive one vector. To complete the test, 247020 clock cycles are needed. In our scheme, the 4-to-20 RPCT interface can receive a vector in five cycles with a 4-bit MVL channel, so that it takes a total of 61755 cycles to complete the test. In both schemes, the number of test pins is reduced from 23 to four. But our scheme has a superior performance in test speed improvement, which is four times the speed of the RPCT-only scheme.

The hardware implementation picture is shown as Figure 9.4. The multiplexer shown in the picture is the CD74HC4067 [7] from Texas Instruments. The ADC

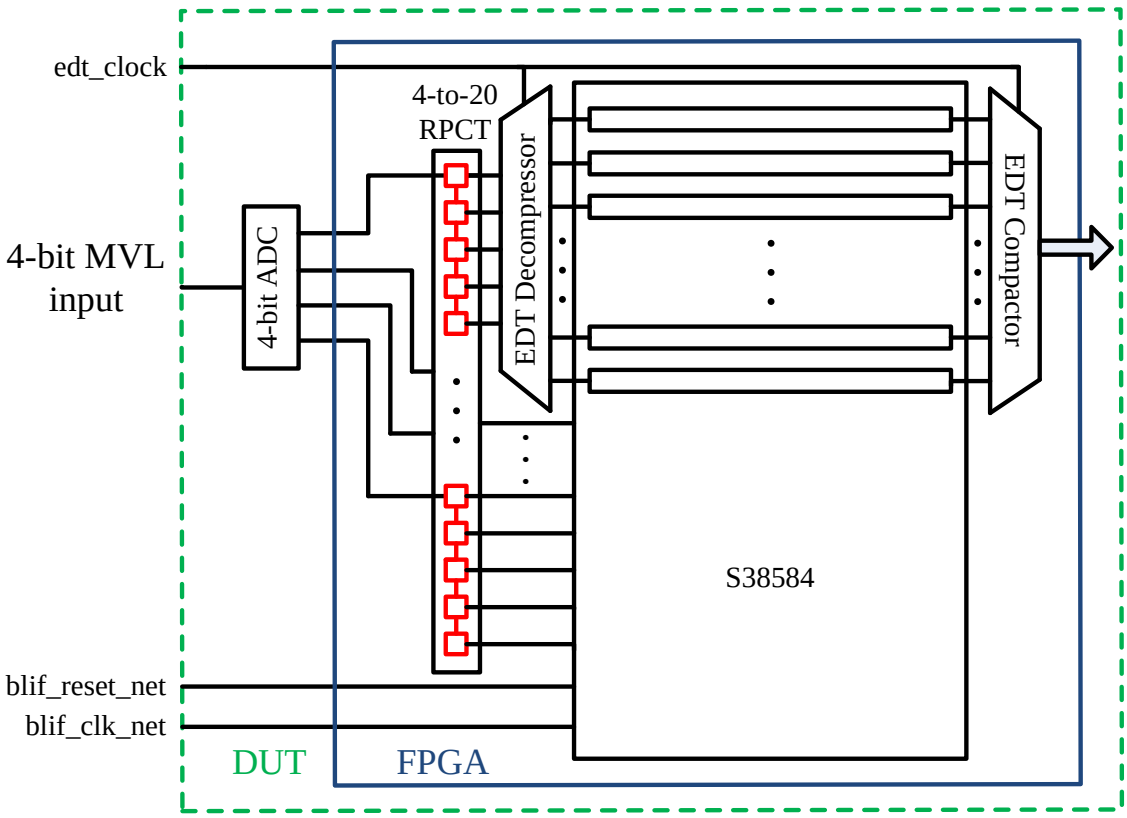


Figure 9.3: Structural view of implemented DUT.

used is the 8-bit AD9057 [4] from Analog Devices, for which we used the first 4 bits of output to work as a 4-bit ADC with larger noise margin. The DUT structure except for the ADC, is programmed into the DE2 FPGA board [5]. The output of the DUT is captured by the ATE through 250MHz DM channels. To test the RPCT-only scheme, we bypass the ADC, directly sending test data to the FPGA board through one 250MHz DM channel. The clock frequency of test channels is 2MHz. We conducted tests with both schemes, whose results are repeatable and identical to the desired test response. The time reduction in our scheme is justified as well. It takes 247020 $cycles \times 500\ ns/cycle = 123.51\ ms$ to test with the RPCT-only scheme. The ADC has a pipeline structure, for which the decoded pattern appears at the third clock cycle. Therefore two extra clock cycles are needed in our scheme. In total, it

takes $(2 + 61755)\ cycles \times 500\ ns/cycle = 30.88\ ms$ to test with our scheme, which is almost one fourth that of the test time with the RPCT-only scheme. The extra two cycles come from the latency of the pipelined ADC. This experimental result justifies the feasibility of our scheme and shows its significant advantage in test time reduction.

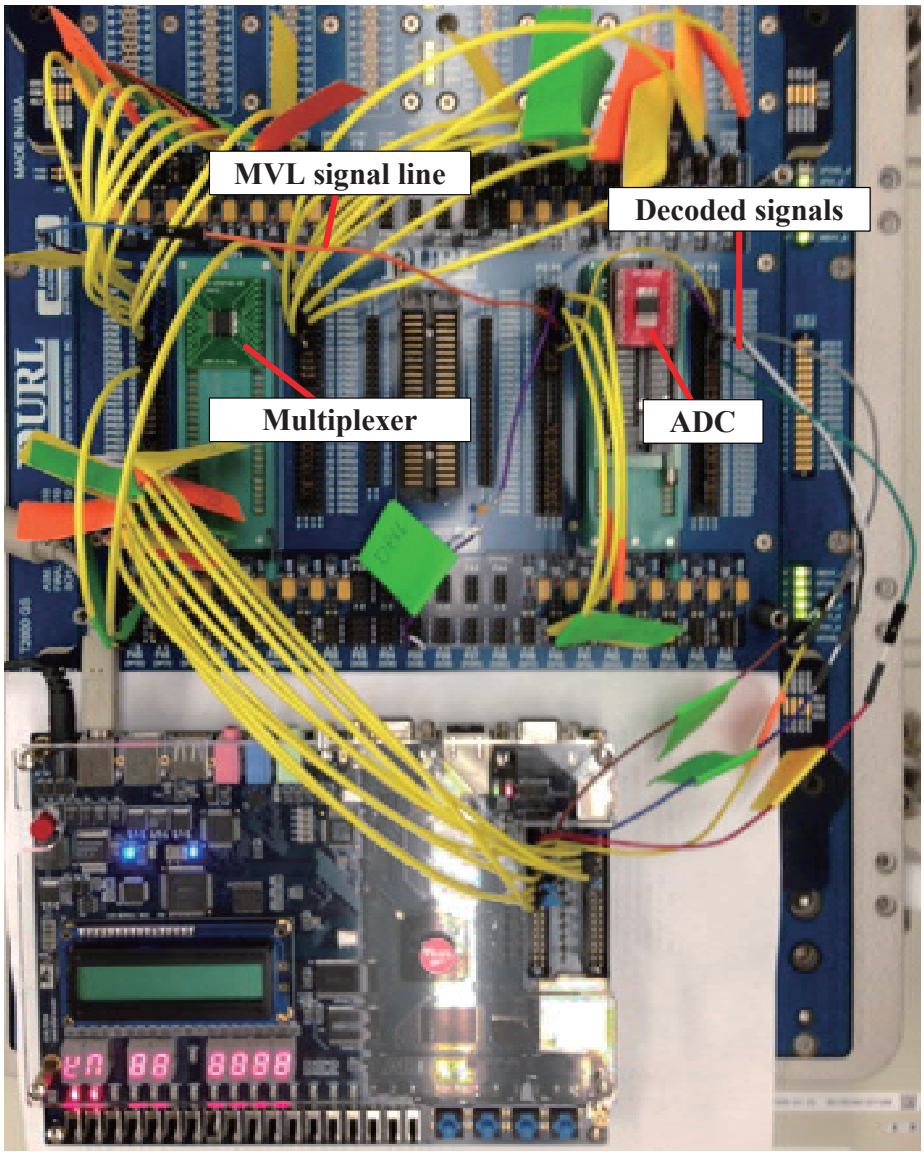


Figure 9.4: Hardware implementation of our ATE based experiment.

Chapter 10

Conclusion

In this thesis, the use of MVL format in test application is proposed, which can increase the data rate of each physical test channel by multiple times. The major challenge in MVL test application is to solve the reliability problem. I analyzed the problem by mathematical models and developed two techniques to tackle with it. A calibration procedure is done before MVL test application to calibrate the nonlinearity of ADC on DUT, which is supported by MVL channel. An error detection and retest based test flow is adopted to ensure the DUT receiving correct test data all the time. Consider about the behavior of MVL test application, we use a signature based detecting method, which reduces the hardware requirement to minimum. According the mathematical analysis, my method will greatly improve the reliability of MVL test application so that possible yield loss and test escape can be avoided.

The benefit of MVL test application may come from test speed improvement or consuming less ATE resource. By replacing every binary channel into MVL channel, the test speed can be increased by multiple times, or to replace multiple binary channels into one MVL channel, keeping the test speed while reducing the number of test pins, which reduces the yield loss in wafer sort and lowers the requirement of test head complexity. However, due to the overhead issue of MVL decoder, it is impractical to send test data to DUT through many MVL channels, which prevents a universally use of MVL test application.

Fortunately, MVL test application is suitable for some current popular test scheme, which is the RPCT with test compression. Now the test compression tools start to support RPCT in their schemes and [27] has talked about the join of test

compression and RPCT in detail. One discovery in [27] is RPCT technology helps increase the input bandwidth of the decompressor with few test channels, which helps the compression algorithm break the input correlation barrier for higher fault coverage. However, we find that the RPCT with test compression scheme will suffer great test speed loss for multi-core SoC designs, in which case the RPCT interface is very large. Then we propose the idea to use MVL test application with RPCT and test compression, which greatly increase the test speed.

To justify the proposed idea, two experiments are done. An ELVIS system based experiment proves the feasibility of MVL test application and the benefit of calibration scheme. In this part, we did SER measurement for data converter pair and conduct scan based test for s298 with MVL test pin. The next experiment is to generate MVL signal with real ATE system T2000GS and justify the idea of combining MVL test application, RPCT and test compression technology. The experiment result shows a big test time saving with our scheme compared to [27].

Bibliography

[1] "7 Series FPGAs Overview, Xilinx," http://www.xilinx.com/support/documentation /data_sheets/ds180_7Series_Overview.pdf, accessed on 04/22/2015.

[2] "AD557: Low Cost, Complete μP-Compatible 8-Bit DAC, ANALOG DEVICES.," www.analog.com/static/imported-files/data_sheets/AD557.pdf, accessed on 04/22/2015.

[3] "AD7822: 3V/5V, 2MSPS, 8-Bit Sampling ADC, ANALOG DEVICES.," www.analog.com/static/imported-files/data_sheets/AD7822_7825_7829.pdf, accessed on 04/22/2015.

[4] "AD9057: 8-Bit 40 MSPS/60 MSPS/80 MSPS A/D Converter, Analog Devices.," http://www.analog.com/media/en/technical-documentation/data-sheets/AD9057.pdf, accessed on 04/22/2015.

[5] "Altera: DE2 Development and Education Board," http://www.altera.com/ education/univ/materials/boards/de2/unv-de2-board.html, accessed on 04/22/2015.

[6] "Cadence: Encounter DFT Architect," http://www.cadence.com/rl/Resources/ datasheets/6083_EncTestArch_DS2.pdf, accessed on 04/22/2015.

[7] "CD74HC4067: High-Speed CMOS Logic 16-Channel Analog Multiplexer/ Demultiplexer, Texas Instruments," http://www.ti.com/lit/ds/symlink/ cd74hc4067.pdf, accessed on 04/22/2015.

[8] "ISCAS89 Sequential Benchmark Circuits," http://www.pld.ttu.ee/ maksim/benchmarks/iscas89/verilog/, accessed on 04/22/2015.

[9] "ITC'99 Benchmark.," http://www.cerc.utexas.edu/itc99-benchmarks/bench .html, accessed on 04/22/2015.

[10] "LabVIEW: System Design Software, National Instruments.," http:// www.ni.com/labview/, accessed on 04/22/2015.

[11] "Mentor Graphics: Tessent TestKompress," http://www.mentor.com/products /silicon-yield/products/testkompress/, accessed on 04/22/2015.

[12] "NI ELVIS: Educational Design and Prototyping Platform, National Instruments.," http://www.ni.com/nielvis/, accessed on 04/22/2015.

[13] "Synopsys: High Quality, Low Cost Test," http://www.synopsys.com/Tools/ Implementation/RTLSynthesis/Test/Documents/dftmax_ds.pdf, accessed on 04/22/2015.

[14] "Advantest R&D Center Inc., OPENSTAR Test Programming Language (OTPL)," 2003. Advantest Corporation technical publication.

[15] "IEEE Standard Testability Method for Embedded Core-based Integrated Circuits," *IEEE Std. 1500*, 2005.

[16] "Test and Test Equipment," in *International Technology Roadmap for Semiconductors 2011 Edition*, 2011. "www.itrs.net/Links/2011ITRS/2011Chapters/2011Test.pdf".

[17] "40- and 100-Gbps Ethernet MAC and PHY MegaCore Function User Guide," 2014. https://www.altera.com/en_US/pdfs/literature/ug/ug_40_100gbe.pdf, accessed on 04/22/2015.

[18] M. Aoyama, K. Ogasawara, M. Sugawara, and T. Ishibashi, "3 Gbps, 5000 ppm spread spectrum SerDes PHY with frequency tracking phase interpolator for serial ATA," in *Proc. Symposium on VLSI Circuits*, 2003, pp. 107–110.

[19] A. Athavale and C. Christensen, *High-Speed Serial I/O Made Simple*. Xilinx, 2005.

[20] B. Murmann, "ADC Performance Survey 1997-2013," http://www.stanford.edu/∼murmann/adcsurvey.html, accessed on 04/22/2015.

[21] S. Bahukudumbi and K. Chakrabarty, "Test-Length and TAM Optimization for Wafer-Level Reduced Pin-Count Testing of Core-Based SoCs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 111–120, 2009.

[22] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the Foundation for Compressed ATPG Vectors," in *Proc. IEEE International Test Conference*, 2001, pp. 748–757.

[23] K. Basu and P. Mishra, "Test Data Compression Using Efficient Bitmask and Dictionary Selection Methods," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 18, no. 9, pp. 1277–1286, Sept. 2010.

[24] C. Berglund, "A Unified Yield Model Incorporating Both Defect and Parametric Effects," *IEEE Trans. on Semiconductor Manufacturing*, vol. 9, no. 3, pp. 447–454, 1996.

[25] R. Budruk, D. Anderson, and T. Shanley, *PCI Express System Architecture*. Addison-Wesley Developer's Press, 2003.

[26] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Springer, 2000.

[27] K. Chakravadhanula, V. Chickermane, D. Pearl, A. Garg, R. Khurana, S. Mukherjee, and P. Nagaraj, "SmartScan - Hierarchical Test Compression for Pin-limited Low Power Designs," in *Proc. IEEE International Test Conference*, 2013. Paper 4.2.

[28] A. Chandra and K. Chakrabarty, "System-on-a-Chip Test-Data Compression and Decompression Architectures Based on Golomb Codes," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 3, pp. 355–368, 2001.

[29] A. Chandra and K. Chakrabarty, "Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (F-DR) Codes," *IEEE Trans. on Computers*, vol. 52, no. 8, pp. 1076–1088, 2003.

[30] A. Chandra, R. Kapur, and Y. Kanzawa, "Scalable Adaptive Scan (SAS)," in *Proc. Design, Automation and Test in Europe Conference and Exhibition*, 2009, pp. 1476–1481.

[31] P.-Y. Chen, C.-W. Wu, and D.-M. Kwai, "On-Chip TSV Testing for 3D IC before Bonding Using Sense Amplification," in *Proc. 18th IEEE Asian Test Symposium*, 2009, pp. 450–455.

[32] R. Chen and S. Chang, "A 6-bit Current-Steering DAC With Compound Current Cells for Both Communication and Rail-to-Rail Voltage-Source Applications," *IEEE Trans. on Circuits and Systems*, vol. 59, no. 2, pp. 746–750, Nov. 2012.

[33] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer, "Low Power Embedded Deterministic Test," in *Proc. 25th IEEE VLSI Test Symposium*, 2007, pp. 75–83.

[34] B. Erkmen, "Integrated Circuit Technology Overview," http://www.yildiz.edu.tr/∼nicoskun/Technology_Overview.pdf, accessed on 04/22/2015.

[35] B. A. Forouzan, *Data Communications and Networking*. McGraw-Hill, 4th edition, 2007.

[36] S. K. Goel and E. J. Marinissen, "Optimisation of on-chip design-for-test infrastructure for maximal multi-site test throughput," in *Proc. Computers and Digital Techs.*, 2005, pp. 442–456.

[37] M. Gustavsson, J. J. Wikner, and N. N. Tan, *CMOS Data Converters for Communications*. Kluwer Academic, 1st edition, 2002.

[38] K. Ichiyama, M. Ishida, K. Nagatani, and T. Watanabe, "A functional test of 2-GHz/4-GHz RF digital communication device using digital tester," in *Proc. IEEE International Test Conference*, 2013. Paper 9.1.

[39] M. Ishida, K. Ichiyama, D. Watanabe, M. Kawabata, and T. Okayasu, "Real-time testing method for 16 Gbps 4-PAM signal interface," in *Proc. IEEE International Test Conference*, 2012. Paper 11.3.

[40] V. Iyengar, S. K. Goel, E. J. Marinissen, and K. Chakrabarty, "Test resource optimization for multi-site testing of SOCs under ATE memory depth constraints," in *Proc. IEEE International Test Conference*, 2002, pp. 1159–1168.

[41] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, "Scan Vector Compression/Decompression Using Statistical Coding," in *Proc. 17th IEEE VLSI Test Symposium*, 1999, pp. 114–120.

[42] M. Jose and H. Wekmann, *An Engineer's Guide to Automated Testing of High-speed Interfaces*. Artech House, 2010.

[43] X. Kavousianos, E. Kalligeros, and D. Nikolos, "Optimal Selective Huffman Coding for Test-Data Compression," *IEEE Trans. on Computers*, vol. 56, no. 8, pp. 1146–1152, 2007.

[44] X. Kavousianos, E. Kalligeros, and D. Nikolos, "Multilevel-Huffman Test-Data Compression for IP Cores with Multiple Scan Chains," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 16, no. 7, pp. 926–931, 2008.

[45] B. Keller, "Encounter Test OPMISR+ On-Chip Compression," in *Proc. IEEE International Test Conference*, 2005. Panel 5.2.

[46] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, C. S. Patel, R. J. Polastre, K. Sakuma, R. Sirdeshmukh, E. J. Sprogis, S. M. Sri-Jayantha, A. M. Stephens, A. W. Topol, C. K. Tsang, B. C. Webb, and S. L. Wright, "Extending OPMISR Beyond 10x Scan Test Efficiency," *IEEE Design and Test of Computers*, vol. 19, no. 5, pp. 553–569, Sept. 2002.

[47] J. U. Knickerbocker, P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, C. S. Patel, R. J. Polastre, K. Sakuma, R. Sirdeshmukh, E. J. Sprogis, S. M. Sri-Jayantha, A. M. Stephens, A. W. Topol, C. K. Tsang, B. C. Webb, and S. L. Wright, "Three-Dimensional Silicon Integration," *IBM Journal of Research and Development*, vol. 52, no. 6, pp. 553–569, Nov. 2008.

[48] J. C. Koob, S. A. Ung, B. F. Cockburn, and D. G. Elliott, "Design and Characterization of a Multilevel DRAM," *IEEE Trans. on Very Large Scale Integration and Systems*, vol. 19, no. 9, pp. 1583–1596, Sept. 2011.

[49] L. Kull, T. Toifl, M. Schmatz, P. Francese, C. Menolfi, M., B. Kossel, T. Morf, T. Andersen, and Y. Leblebici, "A 3.1mW 8b 1.2GS/s Single-Channel Asynchronous SAR ADC with Alternate Comparators for Enhanced Speed in 32nm Digital SOI CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 468–469.

[50] B. Li and V. D. Agrawal, "Multivalued Logic for Reduced Pin Count and Multi-site SoC Testing," in *North Atlantic Test Workshop*, 2015.

[51] B. Li, B. Zhang, and V. D. Agrawal, "Testing With Reduced ATE Channels (Talk)," in *North Atlantic Test Workshop*, 2014. http://www.eng.auburn.edu/ agrawvd/TALKS/NATW14/natw14_Li.pdf, accessed on 04/22/2015.

[52] B. Li, B. Zhang, and V. D. Agrawal, "Adopting Multi-Valued Logic for Reduced Pin-Count Testing," in *Proc. 16th Latin-American Test Symposium*, 2015.

[53] L. Li and K. Chakrabarty, "Test Data Compression Using Dictionaries with Fixed-Length Indices," in *Proc. 21st IEEE VLSI Test Symposium*, 2003, pp. 219–224.

[54] C.-H. Lin and K. Bult, "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm2," *IEEE Journal Solid-State Circuits*, vol. 33, no. 12, pp. 1948–1958, Dec. 1998.

[55] A. M. Majid and D. C. Keezer, "Stretching the Limits of FPGA SerDes for Enhanced ATE Performance," in *Design, Automation and Test in Europe Conference and Exhibition*, 2010, pp. 202–207.

[56] E. J. Marinissen, J. Verbree, and M. Konijnenburg, "A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs," in *Proc. 28th IEEE VLSI Test Symposium*, 2010, pp. 269–274.

[57] MathWorks., "Matlab: the Language of Technical Computing," http://www.mathworks.com/products/matlab/, accessed on 04/22/2015.

[58] A. Mochizuki and T. Hanyu, "A 1.88ns 54×54-bit Multiplier in 0.18μm CMOS Based on Multiple-Valued Differential-Pair Circuitry," in *Symposium on VLSI Circuits Digest of Technical Papers*, 2005, pp. 264–267.

[59] G. E. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965. Also, *Proc. IEEE*, vol. 86, no. 1, pp. 82-85, Jan. 1998.

[60] G. E. Moore, "Progress in Digital Integrated Electronics," *Proc. International Electron Device Meeting*, pp. 11–13, Jan. 1975.

[61] G. E. Moore, "Lithography and the Future of Moore's Law," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 5, pp. 37–42, Sept. 2006. Originally published in *Proc. SPIE*, vol. 2437, pp. 2-17, May 1995.

[62] J. Moreau, T. Droniou, P. Lebourg, and P. Armagnat, "Running Scan Test on Three Pins: Yes We Can!," in *Proc. IEEE International Test Conference*, 2009. Paper 18.1.

[63] J. Moreira, B. Roth, H. Werkmann, M. Klapproth, L. Howieson, M. Broman, W. Ouedraogo, and M. Lin, "An Active Test Fixture Approach for 40 Gbps and Above At-Speed Testing Using a Standard ATE System," in *Proc. 21st IEEE Asian Test Symposium*, 2013, pp. 271–276.

[64] J. Moreira and H. Werkmann, *An Engineer's Guide to Automated Testing of High-Speed Interfaces*. Artech House, 1st edition, 2010.

[65] J. Moreira, H. Werkmann, M. Ishida, B. Roth, V. Filsinger, and S.-X. Yang, "An ATE Based 32 Gbaud PAM-4 At-Speed Characterization and Testing Solution," in *Proc. 23rd IEEE Asian Test Symposium*, 2014, pp. 218–223.

[66] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring Generators - New Devices for Embedded Test Applications," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 9, pp. 1306–1320, Sept. 2004.

[67] B. Murmann, "A/D Converter Circuit and Architecture Design for High-Speed Data Communication," in *Proc. Custom Integrated Circuits Conference*, 2013, pp. 1–78.

[68] T. Nakajima, T. Yaguchi, and H. Sugimura, "An ATE Architecture for Implementing Very High Efficiency Concurrent Testing," in *Proc. IEEE International Test Conference*, 2012. Paper 7.1.

[69] A. Nespola, S. Abrate, R. Gaudino, C. Zerna, B. Offenbeck, and N. Weber, "High-Speed Communications Over Polymer Optical Fibers for In-Building Cabling and Home Networking," *IEEE Journal of Photonics*, vol. 2, no. 3, pp. 347–358, June 2010.

[70] B. Noia, K. Chakrabarty, S. K. Goel, E. J. Marinissen, and J. Verbree, "Test-Architecture Optimization and Test Scheduling for TSV-Based 3-D Stacked ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1705–1718, Nov. 2011.

[71] B. Noia, K. Chakrabarty, and E. J. Marinissen, "Optimization Methods for Post-Bond Die-Internal/External Testing in 3D Stacked ICs," in *Proc. IEEE International Test Conference*, 2010. Paper 6.3.

[72] A. Papanikolaou, D. Soudris, and R. Radojcic, editors, *Three Dimensional System Integration*. Springer, 2011.

[73] G. I. Radulov, P. J. Quinn, and A. H. M. van Roermund, "A 28-nm CMOS 1 V 3.5 GS/s 6-bit DAC With Signal-Independent Delta-I Noise DfT Scheme," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 23, no. 1, pp. 44–53, Jan. 2014.

[74] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded Deterministic Test," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 776–792, May 2004.

[75] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian, "Embedded Deterministic Test for Low Cost Manufacturing Test," in *Proc. IEEE International Test Conference*, 2002, pp. 301–310.

[76] A. Sanghani, B. Yang, K. Natarajan, and C. Liu, "Design and Implementation of A Time-Division Multiplexing Scan Architecture Using Serializer and Deserializer in GPU Chips," in *Proc. 29th IEEE VLSI Test Symposium*, 2011, pp. 219–224.

[77] S.-W. Seong and P. Mishra, "Bitmask-Based Code Compression for Embedded Systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 4, pp. 673–685, 2008.

[78] A. Shirane, H. Tan, Y. Fang, and T. Ibe, "A 5.8GHz RF-powered transceiver with a 113W 32-QAM transmitter employing the IF-based quadrature backscattering technique," in *IEEE International Solid-State Circuits Conference*, 2015, pp. 1–3.

[79] A. Singh, K. Bus, , D. Carnelli, A. Falay, F. Hofstra, K. Licciardello, K. Salimi, H. Santos, A. Shokrollahi, R. Ulrich, C. Walter, J. Fox, P. Hunt, J. Keay,

R. Simpson, A. Stewart, G. Surace, and H. Cronie, "A Pin and Power-Efficient Low-Latency 8-to-12Gb/s/wire 8b8w-Coded SerDes Link for High-Loss Channels in 40nm Technology," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2014, pp. 442–443.

[80] A. D. Singh, "Four Valued Buses for Clocked CMOS VLSI Systems," in *Proc. IEEE International Symposium on Multiple-Valued Logic*, 1987, pp. 285–291.

[81] A. D. Singh, "Precharged CMOS Quarternary Logic Encoder-Decoder Circuits," *International Journal of Electronics*, vol. 67, no. 5, pp. 809–817, 1989.

[82] N. Sitchinava, E. Gizdarski, S. Samaranayake, F. Neuveux, R. Kapur, and T. W. Williams, "Changing the Scan Enable During Shift," in *Proc. 22nd IEEE VLSI Test Symposium*, 2004, pp. 73–78.

[83] B. Song, K. Kim, J. Lee, and J. Burm, "A 0.18-μm CMOS 10-Gb/s Dual-Mode 10-PAM Serial Link Transceiver," *IEEE Trans. on Circuits and Systems*, vol. 60, no. 2, pp. 457–468, Feb. 2013.

[84] U. S. Syed, K. Chakrabarty, A. Chandra, and R. Kapur, "3D-Scalable Adaptive Scan (3D-SAS)," in *Proc. IEEE International 3D Systems Integration Conference*, 2011, pp. 1–6.

[85] S. Tabrizchi, H. Sharifi, F. Sharifi, and K. Navi, "A Novel Design Approach for Ternary Compressor Cells based on CNTFETs," *Journal of Electronic Testing: Theory and Applications*, 2015.

[86] F. N. Taher, "A Low-Power Analog Bus for On-Chip Digital Communication," Master's thesis, Auburn University, Alabama, USA, Aug. 2013.

[87] F. N. Taher, S. Sindia, and V. D. Agrawal, "An Analog Bus for Low-Power On-Chip Digital Communication," in *Proc. 50th Design Automation Conference*, (Austin, TX), June 2013. Poster 61.73.

[88] Y. Takahashi and A. Maeda, "Multi Domain Test: Novel Test Strategy to Reduce the Cost of Test," in *Proc. 29th IEEE VLSI Test Symposium*, 2011, pp. 303–308.

[89] M. Tehranipoor, M. Nourani, K. Arabi, and A. Afzali-Kusha, "Mixed RL-Huffman Encoding for Power Reduction and Data Compression in Scan Test," in *Proc. International Symposium Circuits and Systems*, 2004, pp. 681–684 (Vol. 2).

[90] M. Tehranipoor, M. Nourani, and K. Chakrabarty, "Nine-Coded Compression Technique With Application to Reduced Pin-Count Testing and Flexible On-Chip Decompression," in *Proc. Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 1284–1289 (Vol. 2).

[91] N. A. Touba, "Survey of Test Vector Compression Techniques," *IEEE Design and Test of Computers*, vol. 23, no. 4, pp. 294–303, 2006.

[92] M. Trawka, G. Mrugalski, N. Mukherjee, J. Pogiel, A. Rajski, J. Janicki, and J. Tyszer, "High-Speed Serial Embedded Deterministic Test for System-on-Chip Designs," in *Proc. 23rd IEEE Test Symposium*, 2014, pp. 1–6.

[93] F. Vartziotis, X. Kavousianos, K. Chakrabarty, R. Parekhji, and A. Jain, "Multi-Site Test Optimization for Multi-Vdd SoCs Using Space- and Time-Division Multiplexing," in *Design, Automation and Test in Europe Conference and Exhibition*, 2014, pp. 202–207.

[94] N. Velamati and R. Daasch, "Analytical Model for Multi-site Efficiency with Parallel to Serial Test Times, Yield and Clustering," in *Proc. IEEE 27th VLSI Test Symposium*, 2009, pp. 411–416.

[95] P. Venkataramani, S. Sindia, and V. D. Agrawal, "Finding best voltage and frequency to shorten power-constrained test time," in *Proc. 31th IEEE VLSI Test Symposium*, 2013, pp. 1–6.

[96] J. Verbree, E. J. Marinissen, P. Roussel, and D. Velenis, "Improving Compression Ratio, Area Overhead, and Test Application Time for System-on-a-Chip Test Data Compression/Decompression," in *Proc. Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 604–611.

[97] E. Volkerink, A. Khoche, L. Kamas, J. Rivoir, and H. Kerkhoff, "Tackling Test Trade-offs from Design, Manufacturing to Market using Economic Modeling," in *Proc. IEEE International Test Conference*, 2001, pp. 1098–1107.

[98] E. Volkerink, A. Khoche, J. Rivoir, and K. Hilliges, "Enhanced reduced pin-count test for full-scan design," in *Proc. IEEE 20th VLSI Test Symposium*, 2002, pp. 411–416.

[99] H. Vranken, T. Waayers, H. Fleury, and D. Lelouvier, "Enhanced reduced pin-count test for full-scan design," in *Proc. IEEE International Test Conference*, 2001, pp. 738–747.

[100] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and architectures.* Morgan Kaufmann, 2006.

[101] M. Zargari, M. Terrovitis, S.-M. Jen, and B. J. Kaczynski, "A Single-Chip Dual-Band Tri-Mode CMOS Transceiver for IEEE 802.11a/b/g Wireless LAN," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 12, pp. 2239–2249, Dec. 2004.

[102] B. Zhang, B. Li, and V. D. Agrawal, "Yield Analysis of a Novel Wafer Manipulation Method in 3D Stacking," in *Proc. IEEE International 3D Systems Integration Conference*, 2013, pp. 1–8.