Reducing the Production Cost of Semiconductor Chips Using (Parallel and Concurrent) Testing and Real-Time Monitoring

Qutaiba Khasawneh
Southern Methodist University, qkhasawneh@smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_electrical_etds

Part of the Electrical and Electronics Commons, and the Electronic Devices and Semiconductor Manufacturing Commons

Recommended Citation
https://scholar.smu.edu/engineering_electrical_etds/32

This Dissertation is brought to you for free and open access by the Electrical Engineering at SMU Scholar. It has been accepted for inclusion in Electrical Engineering Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
REDUCING THE PRODUCTION COST OF SEMICONDUCTOR CHIPS USING PARENT (PARALLEL AND CONCURRENT) TESTING AND REAL-TIME MONITORING

Approved by:

____________________________
Dr. Ping Gui
Professor of Electrical and Computer Engineering

____________________________
Dr. Jennifer Dworak
Associate Professor of Electrical and Computer Engineering

____________________________
Dr. Mitch Thornton
Professor of Electrical and Computer Engineering

____________________________
Dr. Mohammad Khodayar
Professor of Electrical and Computer Engineering

____________________________
Dr. Monnie McGee
Associate Professor of Statistical Science

____________________________
Dr. Qing Zhao
Ph.D. of Electrical Engineering

____________________________
Dr. Theodore Manikas
Ph.D., P.E. Clinical Professor
REDUCING THE PRODUCTION COST OF SEMICONDUCTOR CHIPS
USING PARENT (PARALLEL AND CONCURRENT) TESTING
AND REAL-TIME MONITORING

A Dissertation Presented to the Graduate Faculty of the
Lyle School of Engineering
Southern Methodist University
in
Partial Fulfillment of the Requirements
for the degree of
Doctor of Philosophy
with a
Major in Electrical Engineering
by
Qutaiba Khasawneh

B.A., Electrical Engineering, Jordan University of Science and Technology, Jordan
M.S., Computer Engineering, University of Massachusetts Lowell

December 21, 2019
ACKNOWLEDGMENTS

I would like to express my deep gratitude to my research supervisors Dr. Ping and Dr. Jennifer for their guidance, patient, and useful critiques of the research work. I would like to thank Dr. Monnie McGee, Dr. Benjamin Williams and Alan C. Elliott from the Department of Statistical Science for their assistance with data analysis and hypothesis testing. I would like to thank Anand Muthaiah from Tessolve DTS Inc. for the data we used in the real yield monitoring. Also, I am so grateful to my parents Dr. Abd-Alaziz and Bothina for their support.

Finally, I would like to thank my wife May and kids Saif, Sarah, and Ahmed for their love, support, patient and encouragement which makes my work possible.
Reducing the Production cost of semiconductor chips using PaRent (Parallel and Concurrent) Testing and Real-Time Monitoring

Advisor: Dr. Ping Gui
Doctor of Philosophy degree conferred December 21, 2019
Dissertation completed December 21, 2019

The technology revolution introduced many electronic devices that were used in defense, automotive, aerospace, industrial, medical, and consumer products. Consumer electronics changed the semiconductor industry by developing many new challenges for consumer products. One of the main challenges in the consumer product is that it propelled the volume of production to massive production, e.g. hundreds of millions of cell phones are produced yearly. Moreover, wearable technology, tablets, personal laptops, and other portable devices are other examples of massive production. Combined with the overproduction of consumer products, price pressure is another challenge for consumer products.

To support the price and volume of consumer demand pressure, the semiconductor industry has spent a lot of time cutting the cost of the products and reducing the production time. So, time, money and effort are spent reducing the cost and cycle time in every step of the semiconductor production cycle including the design phase, the manufacturing phase, the packaging, and the testing phase. The testing phase has two levels the wafer level and the packaged level. A wafer is defined as a thin slice of semiconductor used to fabricate the Integrated Circuit (IC). Each wafer has a large number of ICs and they are called dies. After slicing the dies and packing them, they are called chips.

Many of the techniques used in the design and fabrication enabled the integration of more devices in the same chips. Moreover, different types of signals can be fabricated in the
same chip. This reduced the cost of the chips, lowered the power consumption, increased the circuit operation speed, enabled more reliable implementation, and reduced physical size. However, this increased the pressure on the fabrication cost, increased the system complexity, increased the verification requirements, and increased the time on the market strain. Moreover, complications in the testing phase increased and the stress of the cost required new techniques in testing. Now, testing is the bottleneck in high volume production, and it counts for a notable share of the chip cost.

Testing has introduced parallel testing which is known as multi-site testing. Testing many units (i.e. physical chips) simultaneously, as in multi-site, saves the test time and increases the throughput of the production. On the other hand, concurrent testing allows for examinations of many blocks within a single chip simultaneously. Concurrent testing reduces the total test by running many blocks inside the chip together. Combining the Parallel test technique with the Concurrent test on mixed-signal chips to have the PaRent testing will amplify the savings in test time and the throughput in production. The PaRent algorithm was developed to address all of the tester’s issue. The cost calculation in the production depends on the tester configuration and the test time of each chip. However, this requires a collaboration between the design team and the test team early in the design process to have the right hooks for the test.

An industrial case study that illustrates the Design-for-Testability (DFT) requirements for successful PaRent testing of a high-volume mixed-signal System-on-Chip (SoC) was published. The published paper outlines the testing obstacles faced while testing a Power-Management Integrated Circuit (PMIC) that was not designed with concurrent testing in mind. Hardware and software optimizations for Automatic Test Equipment (ATE) to improve PaRent testing are described.

This dissertation will also discuss the technique of real-time monitoring of manufacturing yield to be used in the production due to the different variation sources in the production environment. The source of variation could be due to process variations, design marginality, or correspond to actual problems with the devices being tested and should lead to those
devices being scrapped and/or changes being made to the fabrication process. However, in other cases, the test equipment itself could be unreliable or require an alteration in the test procedure, leading to either a bad chip testing as good or good chips failing the test. Thus, a real-time monitor of the yield data and the quality of the test result data is essential for assuring high test quality and for ruling out possible issues arising from the test hardware. The real-time monitoring of the yield can save the chip manufacturer time and cost by finding issues with the testing system in the early stages of production (or as soon as those issues arise), avoiding the scrapping of good devices and preventing the penalty of sending bad units to the customers.

A low-cost algorithm to monitor the yield of multiple test sites was developed and tested against real-time data. The algorithm is capable of being implemented in the test program of a standard tester, and the ability of the approach to provide an early indication of test site problems during wafer test is proved through an industrial case study. The algorithm added less than 1% to the tester time after testing 480 units instead of testing 5532 units. It also flags the site of the issue and the test with the issue.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................................................... x

LIST OF TABLES ............................................................................................ xiv

CHAPTER

1. INTRODUCTION ..................................................................................... 1

2. PaRent Testing ......................................................................................... 9

2.1. Multi-site Testing ................................................................................. 9

2.1.1. Multi-Site Test Challenges ................................................................. 10

2.1.2. Multi-Site Test Advantages ............................................................... 12

2.2. Concurrent Testing .............................................................................. 12

2.2.1. Concurrent test challenges ............................................................... 13

2.2.2. Concurrent test advantages ............................................................... 15

2.3. PaRent Testing .................................................................................... 15

2.3.1. Tester’s Challenges .......................................................................... 16

2.3.2. PaRent steps .................................................................................. 17

3. Case Study using PMIC Device .............................................................. 20

3.1. Introduction ......................................................................................... 20

3.2. PMIC Chip ......................................................................................... 20

3.3. PMIC Test Challenges ........................................................................ 22

3.3.1. Introduction to the PMIC Original Test ........................................... 24

3.4. PMIC Challenge ................................................................................ 25

3.5. PMIC Test Solution Results ............................................................... 25

3.6. PaRent Steps .................................................................................... 26

3.7. PaRent Results ................................................................................ 30
6.2. Real-Time Yield Monitor Conclusion ........................................... 100
6.3. Real-Time Yield Monitor Future Work ........................................ 101

APPENDIX

A. Statistical Background ............................................................... 103
   A.1. Decision Errors ................................................................. 103
   A.2. Pearson’s Chi-squared test ................................................... 103

BIBLIOGRAPHY ................................................................. 110
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The stages of test development.</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>The rule of ten [1].</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Mixed signal ATE modules where AWC is Analog Waveform Capture and AWG is Analog Waveform Generate</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>The single site test time.</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>The multi-site test time. The tester’s resources are fanned out to the different sites and one block is tested at each site.</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>The concurrent test time. Multiple Blocks in the Device Under Test being tested at the same time using the tester’s resources simultaneously.</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>The PaRent Test time. Multiple Blocks and multiple units are tested simultaneously.</td>
<td>16</td>
</tr>
<tr>
<td>2.5</td>
<td>The PaRent Procedure.</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>The PMIC details.</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>PaRent Vs Sequential Time.</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>The Test Time reduction using Multi-site and concurrent testing. The graph shows the time needed to test 8 units sequentially, with multi-site, and with PaRent testing.</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>The Test Flow for PaRent Testing with total test time of 19.64 Second.</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Western Electrical Rules.</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>The yield variables.</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Normal distribution with mean=0 and different variance 1, 2 and 3.</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Normal distribution with different variance and mean=-0.5 which represent negative systemic error.</td>
<td>46</td>
</tr>
<tr>
<td>4.5</td>
<td>Faulty component causes a shift of the data by one standard deviation.</td>
<td>48</td>
</tr>
</tbody>
</table>
4.6 Rout (Ohm) test result with the low yield. High percentage of units failed the 2 ohm upper limit. ................................................................. 54
4.7 Rout (Ohm) test histogram with the low yield. All of the sites have high yield loss................................................................. 55
4.8 Rout (Ohm) test result of a lot with the expected yield. Only a small percentage of units failed the 2 ohm upper limit.............................................. 55
4.9 Rout (Ohm) test histogram of a lot with the expected yield. Only small percentage of the units failed the 2 ohm high limit in this lot....................... 56
4.10 T3 (Reference Current Test) results by site for the lot with issue. One site is failing the test and the other sites have no issues. ....................... 57
4.11 T3 (Reference Current Test) histogram data by site for the lot with issue. All the sites have the data centered around 10.6uA, but site one has many units around 20uA. ............................................. 58
5.1 Real-time yield monitoring algorithm. ........................................... 60
5.2 Dependency Check Pseudo-Code.................................................. 62
5.3 T1 (Output Resistor) Results for all lots. ....................................... 64
5.4 T1 (Output Resistor) Results and histogram for the lot with lower yield. ...... 65
5.5 Dependency Check Function results for T1 using significance level of 0.01. All of the windows show the test results depend on the site used in testing. The exception is window 24 because window 24 has 14 units which less than 240. Thus, the $\chi^2$ is not run and it assumed to be independent.............................................. 67
5.6 Site location for Dependency Check Function results for T1 using significance value of 0.01. All of the windows show the test results depend on the site used in testing due to an issue on the load board affecting all the sites. .............................................................. 68
5.7 Dependency Check Function results for T1 using significance value of 0.05. Window 24 has 14 units which less than 240. The $\chi^2$ is not run, and it assumed to be independent.................................................. 69
5.8 Site location for Dependency Check Function results for T1 using significance value of 0.05. All the sites have an issue in the test result due to an issue on the load board. The location of the largest residual is not same. The problem was on all of the sites due to load board issue. ........... 70
5.9 T2 (Functional Test) results for all lots. ....................................... 71
5.10 T2 (Functional Test) results for a good lot. ................................. 71
5.11 T2 (Functional Test) results for a lot with a lower yield. ................. 72
5.12 Dependency Check Function results for T2 of Lot 3 using significance value of 0.01. Windows 20 and 23 shows that the test depends on the site used to test it. ................................................................. 73
5.13 Site location for Dependency Check Function results for T2 of Lot 3 using significance value of 0.01. Windows 20 and 23 shows that Site 2 has an issue. It has a higher yield loss than the other sites. ......................... 74
5.14 Dependency Check Function results for T2 using significance value of 0.05. Windows 1, 2, 20, 22, and 23 shows that the test depends on the site used to test it. ................................................................. 75
5.15 Site location for Dependency Check Function results for T2 using significance value of 0.05. Windows 1, 2, 20, 22, and 23 shows that Site 2 has an issue. ................................................................. 76
5.16 T3 (Reference Current Test) results for all lots. .............................. 77
5.17 T3 (Reference Current Test) results for good lot. ............................... 78
5.18 T3 (Reference Current Test) results for a lot with issue. ....................... 78
5.19 Dependency Check Function results for T3 using significance value of 0.01. All of the windows show the test results depend on the site used in testing, but window 24 because window 24 has 14 units only which less than 240. The $\chi^2$ is not run and it assumed to be independent. ............... 79
5.20 Site location for Dependency Check Function results for T3 using significance value of 0.01. Site 1 has an issue that caused the yield to be less than the expected yield. Window 24 does not show any site since it does not have enough samples to run $\chi^2$ test. ......................... 80
5.21 Dependency Check Function results for T3 using significance value of 0.05. All of the windows show the test results depend on the site used in testing, but window 24 because window 24 has 14 units only which less than 240. The $\chi^2$ is not run and it assumed to be independent. ............. 81
5.22 Site location for Dependency Check Function results for T3 using significance value of 0.05. Site 1 has an issue that caused the yield to be less than the expected yield. Window 24 does not show any site since it does not have enough samples to run $\chi^2$ test. ......................... 82
5.23 Sensitivity of 8 sites solution for different expected values with a significance value of 0.05 and 0.01 using 100 and 30 samples. ......................... 88
5.24 Sensitivity results of 2-16 sites solution for different expected values with a
significance value of 0.01. .............................................................. 89

5.25 Sensitivity results of 2-16 sites solution for different expected values with a
significance value of 0.05. .............................................................. 90

5.26 Sensitivity of 8 sites solution for different number of samples and a signifi-
cance value of 0.01. .............................................................. 91

5.27 Sensitivity of 8 sites solution for different number of samples and different
expected yield with a significance value of 0.05. ......................... 92

A.1 Chart of $\chi^2$ distributions. ....................................................... 105
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Trim time for different test in the Octal Site.</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Test Time and Throughput for single site, eight sites using sequential and concurrent Testing. The red font represents the PaRent Testing Data</td>
<td>30</td>
</tr>
<tr>
<td>3.3</td>
<td>Tester equipment usage in sequential flow.</td>
<td>34</td>
</tr>
<tr>
<td>3.4</td>
<td>Tester equipment usage in PaRent.</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>T1 observed data for 480 units.</td>
<td>54</td>
</tr>
<tr>
<td>5.1</td>
<td>The calculated residual for the first window with an issue.</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>Contingency Table for the yield of N sites</td>
<td>86</td>
</tr>
<tr>
<td>5.3</td>
<td>Contingency table when site i had different yield than the expected yield.</td>
<td>86</td>
</tr>
<tr>
<td>A.1</td>
<td>Table of error conditions</td>
<td>103</td>
</tr>
<tr>
<td>A.2</td>
<td>Contingency table of two categories and each category has 2 levels with the observed value and the expected value for each level and category.</td>
<td>105</td>
</tr>
<tr>
<td>A.3</td>
<td>Contingency Table of tossing two coins ten times showing the observed values and the expected values.</td>
<td>106</td>
</tr>
<tr>
<td>A.4</td>
<td>$\chi^2$ calculation for coin tossing experiment.</td>
<td>107</td>
</tr>
<tr>
<td>A.5</td>
<td>Critical $\chi^2$ values for significance level of 0.05 and 0.01 for different degree of freedom calculated using Excel.</td>
<td>108</td>
</tr>
</tbody>
</table>
This is dedicated to the soul of my father whose cancer took his life before he could see this day, my mother, my wife, and my kids Saif, Sarah, and Ahmed.
Chapter 1

INTRODUCTION

According to the *New Oxford English Dictionary* [2], “a test is defined as a procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread”. No matter what is produced, no one will buy a product until it is approved to function the way it is specified.

In 1947, Bell Laboratories exhibited the first transistor [3]. In the last seventy years, the semiconductor industry has miniaturized the transistor’s sizes and was able to produce chips with tens of millions of transistors with higher density and higher speed of operation. In fact, the functionality and the complexity of chips have increased enough to include an entire system on a chip (SOC) [4]. Radio-frequency circuitry, mixed-signal components, power management, digital signal processing, memory, and more functionality may all exist on the same chip. For instance, the Apple A12 Bionic processor comes with 6.9 billion transistors on a die 83.27 mm$^2$ in size [5].

The progression in chip performance required improvement in the engineering process in all the aspects manufacturing, including design, fabrication, and testing. While technological advances have increased the complexity of integrated circuits (ICs), market forces have provided significant pressure to reduce chip costs. The improvement in the design and fabrication processes were able to accommodate the cost demand [6]. However, the complexity of the device paired with time to the market and cost pressure required more testing, verification, and evaluation in every step of design and manufacturing. More testing and verification is needed which leads to extra time and cost. Among all costs related to the production of ICs, the chip testing cost is among the hardest to reduce. In fact, it could account for fifty percent of the total cost of manufacturing complex chips [7]. However, quality assurance and testing chips before shipping them to the customer is an indispensable
step in the semiconductor industry to guarantee excellent quality products and customer satisfaction.

The separation between the design and the test had to be bridged to overcome all the challenges in testing and to enable the testing of all the different blocks in the chip in the most cost-effective way. Moreover, the criticality of the testing was acknowledged, and now it is considered in the early stages of design, and Design-For-Test (DFT) was established [8]. DFT is a design technique used to speed up testing and/or enable higher fault coverage by adding new circuitry to the design [9]. The circuitry added to the design helps ensure that the test engineer has the ability to observe and control the different signals in every block in the chip [9]. Moreover, the DFT makes sure that each block can be tested as quickly as possible to keep the cost as low as possible. The different stages of development are shown in 1.1. The pre-Silicon stage is the time before having the chip fabricated in the silicon. Pre-Silicon includes the design, the DFT, and the test development. Post-Silicon starts when the fabrication process is done. At this the time all of the blocks inside the chip are tested and characterised to see if the manufactured version meets the desired specification. When specifications are not met, changes to the design may be needed. Eventually, the design reaches mass production, where every parts must be tested for manufacturing defects.

Testing is crucial in the semiconductor industry, so much so that at each stage of the development cycle, the integrated chip goes through verification, evaluation, or testing to ensure the functionality of the integrated chip (IC) [1]. The cost of a problem grows higher if it propagates to the next stage. The cost of finding a problem in the design using the simulation tools is less expensive than finding the problem after fabricating the design into dies at the wafer level. Also, the cost of finding a defective die at the wafer is less expensive than the cost finding and throwing a die that has been placed in a package.

According to the rule of ten [1], the cost of finding a fault in the Printed Circuit Board (PCB) level is ten times the cost of finding it at the chip level as shown in Figure 1.2. By the same token, the cost of finding a fault on the system level is ten times more than finding it at PCB level, and the cost of finding it in the field is ten times more than finding it at the
The cost of problems appearing in the field is non-trivial. For example, Intel lost $475 million to recall the Pentium processor in 1997 due to the floating point division (FDIV) bug [10], while in 2011, Intel was charged more the $700 million for the Cougar Point chipset problem [11].

Various technologies such as adaptive testing, advanced Design-For-Test (DFT), faster mixed-signal testers, fine-pitch probe cards, and design standards are the future technologies and approaches expected to significantly change the dynamics of the ATE market and enhance growth [12].

The testing in each stage of the development cycle does not eliminate the testing in the next stage. Each stage has its own complexities, challenges, and variations that could produce defects in the final product. Defects could be process defects, material defects, or package defects. At the wafer level, defects could be caused by any of the following: dust particles on the masks or the projection system, incorrect doping levels, mask misalignment, impurities in the wafer material and chemicals, incorrect temperature, and many more. Packaging the chips necessitates slicing the wafer, picking the good units, and then securely
packaging them [4]. Package defects could be caused by seal leaks, contact degradation, bridging between conductors, missing contacts, and many others issue [4].

Due to the complexity of the chips, the testing goes through two stages; the test development stage and production test stage. The test development starts as soon as the design starts. The test development includes the DFT, load board design used on the Automated Test Equipment (ATE), coding the test program, and then the process of debugging the test program on the wafer or the packaged device. Designing the load board and coding the test program is dependent on the DFT because the tester configuration is chosen to guarantee the capability of the tester to test all the blocks in the chip effectively. It is important to know the number, the size, and the type of resources needed to test a single chip. Knowing
the tester’s configuration will determine the test time budget to keep the cost of testing
within the required margins. The test time budget, the DFT, and the tester configuration
will drive the load board design.

Figure 1.3. Mixed signal ATE modules where AWC is Analog Waveform Capture and AWG
is Analog Waveform Generate

The ATE has many varieties of configuration. Figure 1.3 shows the different modules that
the mixed signal tester could have. The cost of the tester depends on the number of modules
and the type of the modules contained in the ATE. For example, the Radio Frequency (RF)
modules are more expensive than the Analog Waveform Generate (AWG), and the Analog
Waveform Capture (AWC) modules. Also, the digital module could have different features
and the cost depends on those features. All of that will affect the final cost of the ATE.

In the development stage, there is a focus on reducing the test time to reduce the cost
of the chips. This effort is shared between the test team and the design team. For example,
the Built-In-Self-Test (BIST) and ATE are utilized to reduce the test time for the digital subsystem as shown in [13], [14], [15], [16] and [17]. The BIST can reduce test time, and it is used to test the embedded memory or logic operations such as sequential and combinational operations at the functional speed. Also, the use of BIST can reduce the number of pins interfaces needed between the ATE and the device under test. Test Data Compression (TDC) is used to compress the output scan data and decompress the input scan. The TDC is discussed in [18], [19], [20] and [21]. BIST, TDC, and scan use extra hardware in the chip design, which adds extra cost and time to the design cycle and the chips’ cost. However, more test coverage is accomplished using the extra hardware in the chip, and less time is spent on the tester.

Other techniques used to reduce the test time per unit include testing many units in parallel. This is called multi-site testing (MST). MST is one of the most efficient ways of test time reduction as discussed in [22]. Moreover, MST takes advantage of the infrastructure needed to store the test data and the test interface facilities such as prober, handler, and test boards, and allows those resources to be used by multiple chips. However, with growing functionality in each chip, the number of pins grows and limits the number of sites that can be tested simultaneously. Therefore, reducing pin count testing (RPCT) was developed to increase the number of sites, (i.e. chips that could be tested simultaneously) as discussed in [21], [23], and [24]. This is needed because the number of sites depends on the number of the test channels and the number of test pins on a chip.

The Joint Test Action Group (JTAG) created the IEEE Standard 1149.1 [25], which specifies the test access port (TAP) that allows interconnections on a board as well as test data registers inside the chip to be accessed and tested using only for pins (TDO, TDO, TCK and TMS). IEEE STD 1500 [26] specifies wrappers that allow better access to embedded cores during test. It operates in concert with IEEE 1149.1.

In addition to testing multiple chips simultaneously, different blocks or cores of the chips can be concurrently tested. These are just some of the approaches that design and test engineers have to deal with to keep the cost of testing under control.
An algorithm to convert a sequential single-site test program to a parallel and concurrent (PaRent) test program is introduced. The algorithm was developed using an industrial case study and a list of the DFT requirements for successful PaRent testing of a high-volume mixed-signal SoC is documented. The thesis describes the testing obstacles faced while testing a Power-Management Integrated Circuit (PMIC) that was not designed with concurrent testing in mind. Hardware and software optimizations for Automatic Test Equipment (ATE) to enhance PaRent testing are suggested.

The cost of test is crucial for determining the cost of the chip. The cost of the testing is calculated as a function of the number of seconds needed to test the chip. In fact, the cost of testing could account for fifty percent of the total cost of manufacturing complex chips [7]. The cost could rise if other equipment is needed to test the chip at low or high temperature. Therefore, a lot of effort is spent in the development stage to shorten the test time per unit. This test time will be used in the next stage of mass production volume. Nevertheless, before going into mass production, the program has to go through checkpoints. Those check points ensure that the test program and hardware are robust, and the results are independent of the hardware used in testing.

In a mass production environment, the test time and the yield are the foremost drivers of the cost. The time spent on the tester, the handler, and the probe is very valuable. This time affects the cost of the chips and the profit margin. It is important to keep the time on those expensive pieces of equipment to a minimum. On the other hand, the yield is defined as the ratio of the number of the good units to the total number of fabricated units. The production yield is one of the fundamental determinants in the cost of the IC chips. A special consideration is allocated to control and observe the yield to make sure it stays within the expected yield.

However, with high volume production, the variation of the process, packaging, and testing could affect the yield, reduce the throughput, and delay the delivery of customer commitment. Real-time monitoring of the yield can save the chip manufacturer time and cost by detecting defects with the testing system in the early stages of production (or as soon
as those issues arise), avoiding the scrapping of good devices and preventing the penalty of sending bad units to the customer.

A low yield could be due to any of the many manufacturing process variables. Many statistical process controls and six-sigma are applied to achieve the expected yield. In [15], wafer sort bitmap data analysis is used for yield analysis and optimization on static random-access memory (SRAM). In [16], hybrid machine learning is used to improve the prediction and control of the yield. In [15] and [16], the yield control and monitoring are used to control the process parameters.

The yield monitoring and control is necessary to avoid low yield. The low yield affects customers’ commitments and increases the manufacturing cost. The increase in the cost is due to the need to fabricate more units. As the number of fabricated units is increased then more equipment is needed to fabricate, package and test the units. All of this comes with a cost added to the chips’ price.

A low-cost algorithm to monitor the yield of multiple test sites is presented. The method is capable of being implemented in the test program of a standard tester, and the ability of the approach to provide early warning of test site problems during wafer test is demonstrated through an industrial case study. The algorithm can be used in real-time and it can be used in the wafer testing and the packaged chips testing. The other algorithm are post-processing and it is not real time.
The trend in the semiconductor industry today is to lower the cost of the electronics by including many different IP blocks in the same chip, which means that there are more blocks to test. However, there is also more pressure to keep the cost low. The Parallel (multi-site) test technique decreases the test cost by running the test on multiple units at the same time. On the other hand, concurrent testing analyzes many blocks in the chip simultaneously. Combining the Parallel test technique with the Concurrent test on mixed-signal chips to have the PaRent testing will amplify the savings in test time and the throughput in production [27]. PaRent testing unites both techniques to maximize the saving in test time and to reduce the cost of testing.

2.1. Multi-site Testing

In the mid-nineties, the ATE cost started increasing rapidly, and the multi-site approach became widely used for SOC chips tested with high-cost ATE. MST is based on testing more than one die/device at the same time. Figure 2.1 shows the single site test time and Figure 2.2 shows the move to multi-site test. The multi-site test technique increased the testing throughput and reduced the cost of manufacturing the chips. A technical cost model of the different approaches to reduce the test cost was studied in [22].

According to [22], multi-site strategies were the most effective cost-saving techniques for testing chips. The chips studied were of various complexities, including a Set-Top-Box chip (which was a medium complexity design), a microprocessor (which was a high complexity design), and two other devices with the simplest complexity designs.

Figure 2.1 shows the single site test time T per unit. The test time after moving to a multi-site test is shown in Figure 2.2. The total time for the multi-site test will increase by
x% for all the units. The value of x depends on the number of sites and the efficiency of the multi-site code. Moreover, the test time per unit is calculated using the total test time divided by the number of sites. In other words, the test time is inversely proportional to the number of sites.

2.1.1. Multi-Site Test Challenges

However, the transition to multi-site test included many challenges, such as the effect of the testers’ memory size limitation on the ability to load all the digital patterns, ineffective test scheduling, minimizing the total number of Test Access Mechanisms (TAMs), and the optimization of the TAM design [28]. New design techniques that deal with the memory
limitation using a reconfigurable scan chain were introduced in [29].

Laying out many sites on the load board was one of the issues the test engineers had to deal with to successfully implement MST. The problem involved the task of successfully dividing the area of the load board that is controlled by the tester head among all the sites so that each chip would experience an equivalent testing environment. Low variation among the various sites, including the length, the width, and the layout of the traces that serve each site, which can affect the test measurements, is necessary so that each chip is tested under similar circumstances. In [30], a high-range delta-sigma modulator was tested using six sites. The paper proved that considering both the layout of the sites on the load board as well as the software used during the test are imperative to have a reliable solution for mass production multi-site test. Also, [30] has a guideline for laying out the ground for the analog and the digital signals and connecting the relays into the different layers. Finally, those guidelines can be used in designing multi-site load boards.

Figure 2.2. The multi-site test time. The tester’s resources are fanned out to the different sites and one block is tested at each site.
2.1.2. Multi-Site Test Advantages

With the multi-site solution, the total test time in particular chip might go up a little bit due to overhead in the data log, indexing of the different units and site numbers and coding the extra equipment for the other sites. The test time for a single site and multi-site is shown in Figure 2.1 and 2.2. However, the multi-site solution reduces the test time per unit. The reduction depends on the number of sites. If the number of sites is $N$, the test time per unit will be the total test time divided by $N$. Also, the throughput will increase by approximately a factor of $N$. So, the total number of units tested using multi-site per hour is almost $N$ times more than single site solution.

Finally, the higher the throughput, the smaller the number of pieces of capital equipment (such as the testers, the handlers, the probers and the load boards) needed to satisfy the customer’s demands. This will reduce the capital cost needed to produce the chip.

2.2. Concurrent Testing

Concurrent testing reduces the test cost by testing many blocks in the chip simultaneously. Concurrent testing increases the throughput as a consequence of curtailing test time.

A concurrent built-in self-test (BIST) was used to test Random Access Memory (RAM) and Read Only Memory (ROM) on a chip in [31]. The IEEE Standard 1687 was explicitly designed to provide efficient access to embedded instruments, such as BIST engines, through the test port [32]. An embedded instrument is defined in [32] as additional logic or a circuit added to the chip to enable test, debug, process monitoring, or functional configuration. The embedded instrument network could be a standardized network such as a 1149.1, 1500, or 1687 network. Using a 1687 network has advantages over 1500 and 1149.1, as explained in [32]. Also, the use of an external FPGA could be used on the tester or the test board to enable IC testing. Moreover, 1687 helps enable the reuse of the embedded instrument across designs and the ability to operate and schedule instrument concurrently and at runtime.
Concurrent testing was also used to test the LEON SOC as a case study in [33], where the test time and the area cost were quantified. The LEON SOC chip has a 32-bit Central Processing unit (CPU), Advanced Microcontroller Bus Architecture (AMBA) bus and several embedded controls. The paper shows that the design needs to have a wrapper around the cores to isolate each core from the environment around it and to insert a test access mechanism (TAM) to facilitate moving the patterns and responses between the cores to the tester’s pins. Multi-objective scheduling must be done to maximize the number of cores tested concurrently and maximize the throughput of the TAM. Also, the power consumption has to be maintained under certain limits. Moreover, minimizing the test length is a must. The total saving was 55.43% in test time.

In [34], testing identical cores simultaneously was proposed to reduce test time. Exceeding power consumption due to running multiple cores at the same time could cause damage to the chip or fault failures. In response, [35] proposed a technique to reduce power consumption using a gating scheme during the shift process in the scan test. In [36], the test time is calculated using the number of vectors in the pattern set used to test the digital core and the speed of running each pattern. The proposed technique determines the bottlenecks in the test solution in order to modify them as needed.

2.2.1. Concurrent test challenges

Concurrent testing faces many challenges and difficulties starting at the design, the load board design, tester resource assignment, and power management. Power management is needed to balance the chip’s demands with the tester resource’s capabilities. Test scheduling is another challenge where the total test time is minimized while avoiding any tester’s or chip’s resource constraints while testing the different blocks concurrently. Optimizing the time needed to design the chip, the DFT, and the test solution must be done in the early stage of the chip design.

DFT has to deal with more process variation in the semiconductor process as the number of transistors has increase with shrinking feature size inside the chips. Trim was added to
overcome the process variation in some designs. So, what exactly is trim? Trim refers to a few techniques used in the semiconductor industry to achieve a high level of precision. Those techniques can be used at the wafer level or at the package level. The method depends on adjusting some of the circuit elements, such as the resistor in the input stage, to correct the offset voltage of an amplifier. The trim starts by enabling the trim mode. Enabling the trim mode makes the internal node observable and accessible. A digital signal will be implemented to adjust the output of the node. The digital signal modification will stop when the output signal is the closest to the target. The trim circuit will be disabled after completing the trim. The adjustment of the circuit will then be permanent. Trim algorithms are one of the hardest to test concurrently. Trim is a search process that has a minimum
and maximum test time. Running multiple blocks in trim mode could cause difficulties in getting the tests synced at the end of the test and the beginning of the next one.

2.2.2. Concurrent test advantages

Concurrent test can save up to 55.43% of test time as shown in [33]. This is an immense saving in the tester time. This great savings was due to the planning in the design process and having the right DFT in the design. Figure 2.3 shows that the concurrent test will cut the test time by a certain percentage. The savings will depend on the slowest test and the number of blocks that can be tested concurrently. The more blocks tested concurrently; the more savings would occur. According to [37], the concurrent test has the potential to continue to emerge with Moore’s Law. Concurrent testing gives a better prediction about the test cost. Moreover, concurrent testing mimics the real usage of the chip in the final application where more than one block will be running at the same time.

Concurrent testing will increase the throughput due to the reduction in the test time. Also, the shorter the test time the smaller number of capital equipment needed to satisfy the customer’s demands.

Finally, concurrent testing will have a higher utilization of the tester’s equipment. Expensive and low-cost instruments will be used simultaneously. So, the total cost will be averaged out, and the test time will be shorter, which means lower chip cost.

2.3. PaRent Testing

PaRent testing combines the advantages of parallel testing (multi-site testing) and concurrent testing. Testing the mixed signal and power management has used MST for many years. The multi-site technique is a well understood mature technique and well established in the SOC chips. However, most of the concurrent testing in the mixed signal and power management chips were used to test digital cores. Running the analog blocks concurrently is more challenging. There are no digital victors with a specified size and speed. So, the test time can be estimated, but it is not accurate or reliable. Moreover, the layout of the
Figure 2.4. The PaRent Test time. Multiple Blocks and multiple units are tested simultaneously.

load board will influence the settling time. Therefore, the wait time will need some tweaking to get stable, repeatable, and reliable measurements. There is no reward without hard work. Consolidating MST with the concurrent testing imposes immense pressure on both the design engineer and the test engineer.

2.3.1. Tester’s Challenges

The tester also has many limitations relater to MST and concurrent testing. The limitations are the power supply capability limitations, the number of resources available, and the types of resources available in the testers. Power supplies are essential in testing the semiconductor chips, where the digital cores are power hungry. The tester has to accommodate the chip’s needs to ensure the performance of the chip. Also, power supplies are essential for the Power Management Integrated Chip (PMIC) devices, where they are needed as supply and load. Designing a load board for parallel testing is a challenge because the budget of
power supplies and the program flow need to be tracked carefully. Each test in the program flow should not exceed the tester budget at any time when testing the chips in parallel or in concurrent mode. Also, many resources have to be shared among different blocks because those resources have the ability to provide the current needed for testing especially power-hungry blocks, such as the Switched Mode Power Supply (SMPS) blocks, which makes it a challenging task for concurrent testing.

Another challenge for concurrent testing is the software. Usually, each block has its own supplies and loads. Those loads and supplies have their own programming code in addition to their own setup and settling time. Most of the tester’s software gives the user the ability to use one command to program the same type of supplies if they are used in parallel. However, not all of them give the user the capability to program different types of instruments under the same command, especially if they are not programmed to the same voltage or current values. This limits concurrent testing.

2.3.2. PaRent steps

PaRent testing requires a lot of planning and investigation. The planning with IEEE 1500 should start early in the design process to access the different cores in the chip and isolate them. IEEE 1450 helps define the standard test interface language (STIL), and IEEE 1450.6 helps define core test language (CTL). They provide a solution for test information exchange. Also, the test access mechanism (TAM) has to be defined in the design. Moreover, the DFT has to speed up the slow clocks in the analog side of the chip and reduce any long wait times. The design and the DFT could isolate the different cores.

However, supporting a multi-site solution will drive multiple cores in the chip to share one tester’s power supply. This will require the test engineer to make sure that running those cores concurrently will not exceed the tester’s power limit and will not affect the chip performance. Therefore, the same test chip has to be tested sequentially, and then concurrently to make sure that the results are the same for all the cores. This is an essential step to ensure that the tester’s resources are not limiting the chip performance during concurrent
test. This is the first step in the flowchart in Figure 2.5.

Figure 2.5. The PaRent Procedure.

In analog testing, it is possible to have one register controlling more than one block in the chip. Programming this kind of register in sequential flow will affect only those blocks. Nevertheless, going to concurrent testing requires different register settings. The register setups for any block going to run concurrently has to be inspected to make sure that the setting for the bits in that register is right for all the affected blocks inside the chip. This
related to the second step in the flowchart in 2.5.

Finding the right tests to include in PaRent has to be debugged on the tester. Planning is not enough due to the sensitivity of the analog blocks to the external factors such as load board layout and the external components installed on the board. The different components on the board have different tolerance values. Tweaking the wait time allows all sites to have the same test results when testing the same unit and makes sure that the MST is stable across all sites.

The concurrent test requires finding the test time, the instruments used, the instruments’ setup and chip’s setup for every test in the test flow. All of this information is needed to be able to find which tests can be applied concurrently.

Some of the mixed signal tests search for the right voltage setup (like the input low voltage (Vil) and the input high voltage (Vih)) or the right register setup (like the trim test). The trim test usually has more steps to be tested, and the number of steps needed to reach the target value depends on the chip. The slowest and the fastest test time for the trim have to be found to be able to schedule the different tests to run concurrently.

The algorithm makes sure that the test with the slowest time running at the highest speed has no conflict with the other tests’ resources, and it can encapsulate the other test running at the slowest speed. Figure 2.5 shows the steps needed to collect the test time, tester’s resource and device’s setups running the test program using single site, then the multi-site with trimmed and untrimmed units. These steps ensure getting the minimum and the maximum values for test time. Also, the tester’s instruments and devices’ setup are collected to avoid any conflict in the tests.

The algorithm was used to reduce the test time in a high volume PMIC device. The device challenges and the test results are discussed in the next chapter.
Chapter 3
Case Study using PMIC Device

3.1. Introduction

The Power Management Integrated Chip (PMIC) is a high-volume device with a high
cost for testing. The chip has blocks that need a special trimming algorithm and another
block with long settling time combined with a different trim. A need to reduce the cost of
testing was essential to keep the total cost within the targeted budget. The PaRent test was
used to reduce the test cost for the PMIC. Developing PaRent testing for the complex analog
chip demonstrated the challenges, the opportunities, and the steps needed to implement the
PaRent test in this mixed signal chip.

3.2. PMIC Chip

This chip has a thirteen low drop out voltage (LDO), six Switched Mode Power Supply
(SMPS), three external 10 bits Analog to Digital Converter (ADC), Battery thermal moni-
toring ADC, Open Circuit Voltage (OCV), battery measurement ADC, one general purpose
ADC, a coulomb counter (CC) for gas gauging with a 13 bit resolution, programmable boot
control and an I2C controller, as shown in Figure 3.1.

The PMIC utilizes a One Time Programmable (OTP) and an Electrically Erasable Pro-
grammable Read-Only Memory (EEPROM) to program the power-up sequence state ma-
chine to serve different platforms.

The Coulomb counter tracks the coulombs (current integrated over time in and out of
the battery pack). It has a long sampling time reaching 250 milliseconds per sample. The
sampling time has to be long to get as accurate measurements as possible because it is
integrated over time. The CC has to be trimmed, where 8 samples are needed to get an
Figure 3.1. The PMIC details.
accurate reading for the CC.

The PMIC also utilizes hardware and a software battery gauging solution using a 4 channel 10-bit Successive-Approximation-Register (SAR) ADC. The ADCs have to be trimmed using the fit line to set the zero-scale error and full-scale error for the ADC. The algorithm for the best fit line needs at least 10 samples to calculate usable range in the ADC for better accuracy.

The SMPSs and the LDOs have programmable output voltage. It is important to test different output levels on the LDO and the SMPS. It also has a temperature warning and shutdown where high temperature will trigger the shutdown sequence. Many of these blocks have to be trimmed.

Variation in the manufacturing process could impact chip performance. Overcoming the process variation requires extra circuits to be added to the chips. However, digital bits are needed to manage the additional circuits. The extra circuits and extra digital bits are used in the trim procedure. The trim depends on adjusting the digital bits' setting to tune the measured output and get it as close as possible to the desired value. The EEPROM will store the values of the bits with the most accurate measured output value. The process of tuning and storing the digital bit values is called trimming. Trimming consumes a lot of test time because it is a searching process. After programming the EEPROM, the chip has to be rebooted, and the output of the circuit is measured again. This will confirm that the EEPROM and the circuit continued the right values. This will add more time to the test time. Therefore, the extra cost is added to the original test cost. The PMIC device has 16 different outputs to be trimmed to achieve higher accuracy.

Keeping the cost down requires fewer blocks to trim or a fast trim algorithm. However, the trend in the industry is to add more bits to the trim blocks to get higher accuracy. Those extra bits will require more time to run the trim algorithms. Trimming concurrently is another way to trim the blocks if the design allows trimming multiple blocks at the same time and if their outputs are accessible at different test points (output pins).
3.3. PMIC Test Challenges

The test solution was first done in the classic way, where the tests were tested sequentially. The first step after powering the chips is programming the default values in the EEPROM to set the right power up sequence for the device. Also, the trim will run on the following block values:

- The internal voltages.
- The real time clock oscillator.
- The internal reference currents.
- The output voltages for the 6 SMPSs.
- The output of three LDOs.
- The CC.
- The ADCs have to be trimmed and calibrated.

The challenge in cutting the test time for this chip lies in the CC. The CC monitors the charging and discharging current of the battery during specific times with a total run time of 250 milliseconds. Moreover, an internal offset calibration is needed to minimize the offset error for the CC.

To test the offset error on the ADC and the CC, the inputs are shorted together to calibrate the ADC/CC in order to obtain the offset voltage. The process of internal calibration needs at least two seconds, where 8 samples are taken. The offset is the average of the eight samples.

The Electrically Erasable Programmable Read-Only Memory (EEPROM) will store the calibration values. The customer will have access to the stored offset value in the EEPROM, and the values can be used in the final application.

Furthermore, other ADC channels need to be calibrated. The ADC calibration algorithm demands 10 points for the best fit line to calculate the usable range of the ADC for better
accuracy. Additionally, the sequence of the trimming blocks must be kept as required by the
design. The reference voltages, currents, and clock must be trimmed before other blocks are
trimmed. Although the other blocks need to be calibrated and tested, none of them need
as long a wait time as the CC. All the calibrated and trimmed blocks have to be verified to
ensure that they hold the programmed values in the EEPROM. Also, the outputs of the CC
are on the expected values. All these consideration add to the test time.

The SMPS in the device can support loads from one ampere up to three amperes. This
limits testing the SMPS concurrently, and it also limits the number of units that can be tested
in parallel at maximum load. Furthermore, testing high currents in production can cause
damage for the test board and the components in the board. Therefore, the program was
designed to test 5% of the max load at production and maximum load at characterization.
For the LDO, a test net is needed in order to be able to measure the output voltage of the
device without the effect of the impedance in the socket and the load board trace. This means
that thirteen test nets are needed in order to be able to test all of the LDOs concurrently.
However, the chip has only one test net, and it is multiplexed to the thirteen LDO output
voltages. This will prevent testing the LDO concurrently with any load. Nevertheless, they
can be tested concurrently if there is no load.

3.3.1. Introduction to the PMIC Original Test

The test cost was expected to be high due to the complexity of the chip; therefore, the
test solution was designed to have an eight site solution. However, testing eight units in
parallel was not enough to keep the test cost within the budget. The need to calibrate the
Analog to Digital Converters (ADCs) and the CC drove the test cost beyond the projected
budget for the chip.

Nevertheless, due to time to market pressure, no DFT was done to test the different blocks
concurrently. Many other tricks were used to reduce the test time, but it was not enough to
keep the cost within a reasonable margin. The only solution was to check the possibility of
running the CC and ADC concurrently. Unfortunately, the DFT and the load board were
not designed to accommodate the concurrent testing due to the time to market pressure. Thus, implementing the concurrent testing required more time and more debugging.

3.4. PMIC Challenge

Having thirteen power supplies in the tester for each module in the chip is extremely difficult. It is more difficult when five blocks are DC-DC converters and two of them are high current blocks. The DC-DC converters are two Amperes DC-DC converters. The DC-DC converter needs two power supplies: one as a source and the other supply as the load. Going for an eight-site solution requires sharing some of the testers’ supply between the different blocks in the chip. Sharing a power supply between different blocks makes it harder to measure the quiescent current (Iddq) of each block. Designing the eight site solution forced using the two power supplies for the five DC-DC converters as a source and as a load. However, more relays on the test board are added to connect only one DC-DC converter to the load.

Moreover, five blocks with the same voltage domain are connected to one power supply in the tester. For the remaining three blocks, each has one power supply; therefore, each site has six power supplies (five as a source and one as load). A multiplexer is used to connect the three ADC channels to the tester resources. The right digital channels in the tester are selected to enable the scan, scan clock, scan input and scan output signals of the scan chains.

3.5. PMIC Test Solution Results

Testing one site requires 19.8 seconds, while using a multi-site solution with 8 sites is 21.71 seconds. The effective test time of a unit in the multi-site solution is 2.71 seconds. The test time is 13.7% of the single test time, however, the test time of 2.71 is above the budget of the project. The budget for test time is 2.5 seconds at most, making the single-site test time 8.5% above the budget.

Table 3.1 shows the test time for the trim of some of the blocks without the post-trim test. The table has the data for the tests that can run concurrently. The multi-site solution needs
12 seconds to trim all of the blocks and to run the post-trim test, including the programming of the EEPROM default values, trimming 4 different internal voltage references, the RTC oscillator, internal reference current, the output of the LDOs, output voltage for 6 SMPS and the internal clock for one of them. For all of the above trims, the wait time was optimized; numbers of samples were reduced to the minimum to keep a good gauge repeatability and reproducibility (GRR).

Moreover, a fast trim procedure was used where possible. However, the ADC and CC trim and post-trim tests take up as much as 9.9 seconds of the 12 seconds. The 9.9 seconds are used for a 4 channel 10 bit SAR ADC and a 13 bit $\Sigma\Delta$ ADC with stringent accuracy and linearity specifications and approximately 46 EEPROM bits allocated. The bottleneck is the CC which has a slow conversion time of 250 milliseconds. While fast conversion modes were provided in DFT modes to achieve up to a 16X reduction in conversion time, the bit accuracy was compromised and found unsuitable for the trim at the rated 13-bit resolution level. The simplest 2 points gain error calibration requires at least 250 milliseconds x 2 = 500 milliseconds. Offset self-calibration takes up 250 milliseconds x 8 samples = 2 sec.

In contrast, each of the other voltage regulators and reference trims take on average 150-250 milliseconds. If performed sequentially, the naive approach would be to initiate, for example, a CC self-calibration, wait 2 seconds, and then read back offset values. Interleaved trim, as an example, eliminates the wait and stuffs into the same period, other trim routines that do not impact the CC and are launched and completed by the time the IC is ready with CC results.

Figure 3.2 shows that using PaRent testing that we developed and discussed in [27] could save time by running many blocks at the same time. Moreover, in real application, many blocks will be working at the same time. Using the PaRent testing will resemble real-time usage of the chip and it will save cost.

3.6. PaRent Steps

PaRent cannot be done arbitrarily. Next, we discuss the iterative steps followed to arrive
at the optimal choice of interleaved trim sequences. In brief, they are stated below:

1. The different blocks to be tested concurrently must be isolated in the design.

2. The order of trimming the different blocks must be set by the designer and kept in the test program.

3. The isolation of the different testers’ instruments must be guaranteed by the load board design.

4. The shortest test time for each trimmed block is the multi-site solution. This is done by running a single site with good units where the fast trim procedure runs with one step.

5. The ability to switch between single site test and multi-site test is needed because it is easier to debug the program in a single site for both serial and concurrent flows.

6. The ability to switch between serial flow and concurrent flow in the case of debugging the program and customer returns is needed.
The trim, as we said before, is a searching algorithm to find the right setting for the trim register to get the specified output value. Therefore, the test time for it fluctuates. The hurdle is to find the quickest and the slowest test time for each block. The slowest time for the test that encapsulates other tests should be larger than the sum of the test time of the encapsulated tests running at the slowest test time. Trimming a trimmed unit will require the shortest and fastest test time. Finding the slowest time can be done by running the trim as a linear search and finding the trimming values for all the steps in the trim. Having the slowest test encapsulate the fastest test time of the other tests will ensure that the test’s results are not interrupted. This will keep the high quality of the tests.

The designers checked and affirmed that the SMPS, the LDOS and CC blocks are isolated and can be tested simultaneously. Moreover, trimming the SMPSs and the LDOs concurrently with the CC will not affect the trimming and quality of the chip. The next step is checking the tester’s instrument isolation. After reviewing the hardware design, it was clear that the CC, the five Bucks (a type of SMPS), the LDOs, and the ADC are not sharing any of the power supplies. Theoretically, concurrent testing is possible between the CC and the other blocks. However, the layout could cause interference between the different power supplies. It is important to run and compare the test results of all the sites when the test program runs sequentially and concurrently.

GRR has to be run to make sure that the concurrent testing did not affect the test quality and that no register is being used by different tests at the same time with different values. So, reading and comparing the chip’s register in each test is done to ensure that.

Also, the shared power supplies have to be checked to make sure that none of the testers’ hardware will exceed the hardware limits and capability. This will guarantee the quality of the test’s results. Having numerous blocks tested at the same time and sharing the tester’s supplies should not affect the test results. The test setups should not conflict with each other. The register setup must be checked to confirm that there is no conflict in the test setup.

Table 3.1 has all the tests in the trim that can be tested concurrently. The other blocks
Table 3.1. Trim time for different test in the Octal Site.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Test Time (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIM_CC</td>
<td>3396.06</td>
</tr>
<tr>
<td>TRIM_Buck1</td>
<td>155.4</td>
</tr>
<tr>
<td>TRIM_Buck2</td>
<td>169</td>
</tr>
<tr>
<td>TRIM_Buck3</td>
<td>179.48</td>
</tr>
<tr>
<td>TRIM_Buck4</td>
<td>165.02</td>
</tr>
<tr>
<td>TRIM_Buck5</td>
<td>183.05</td>
</tr>
<tr>
<td>TRIM_Buck6</td>
<td>166.26</td>
</tr>
<tr>
<td>TRIM_LDO</td>
<td>233.05</td>
</tr>
<tr>
<td>TRIM_ADC</td>
<td>722.25</td>
</tr>
</tbody>
</table>

Total Time of test in ADC, BUCK and LDO: 1973.51

Total Time of test in ADC, BUCK, LDO and CC: 4793.57
not in the table have to be tested in a certain order due to design requirements.

3.7. PaRent Results

The result of implementing the concurrent test on the original solution is a cut in the test time. The test savings are shown in Figure 3.3 and the exact numbers of saving are shown in Table 3.2.

Figure 3.3. The Test Time reduction using Multi-site and concurrent testing. The graph shows the time needed to test 8 units sequentially, with multi-site, and with PaRent testing.

Table 3.2. Test Time and Throughput for single site, eight sites using sequential and concurrent Testing. The red font represents the PaRent Testing Data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sites</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Sequential Test Time (S)</td>
<td>19.18</td>
<td>21.7</td>
</tr>
<tr>
<td># of units per hour</td>
<td>188</td>
<td>1327</td>
</tr>
<tr>
<td>Concurrent Test Time (S)</td>
<td>17.23</td>
<td><strong>19.64</strong></td>
</tr>
<tr>
<td># of units per hour</td>
<td>209</td>
<td><strong>1466</strong></td>
</tr>
</tbody>
</table>

Table 3.2 has the test time data for running the program using one site, eight sites, sequential test, and concurrent test. The table also shows the throughput of different configurations. The test time always increases as the number of sites increases. The extra time
is due to overhead in programming the extra instrument for the other site, logging the test results and sorting the test results. The throughput of increased to 1327 units per hour from 188 units per hour when MST is used, concurrent testing with single site reduced the test time from 19.18 seconds to 17.23 seconds. This is an 11% saving in test time and increases the throughput per tester. On the other hand, combining the concurrent test with the multi-site test (PaRent test) reduced the test time from 21.7 seconds to 19.64 seconds. The throughput with PaRent test is 1466 units per hour per tester.

3.8. PaRent Advantages

PaRent testing can speed up the test time and throughput. The speedup in test time can be calculated using the equation is

\[
\text{Speedup per unit} = \frac{\text{Test Time Before}}{\text{Test Time After}} \times 100\%.
\]

\[
\text{Speedup per unit(single Site Sequential to 8 sites Sequential)} = \frac{19.18}{21.7/8} \times 100\% = 707\%
\]

\[
\text{Speedup per unit(single Site Sequential to single concurrent)} = \frac{19.18}{17.23} \times 100\% = 111\%
\]

\[
\text{Speedup per unit(8 Sites Sequential to 8 sites concurrent)} = \frac{21.7/8}{19.64/8} \times 100\% = 110\%
\]

\[
\text{Speedup per unit(single to PaRent)} = \frac{19.18}{19.64/8} \times 100\% = 781\%
\]

The PaRent test enhances the utilization of the tester's different equipment. Table 3.3
Figure 3.4. The Test Flow for PaRent Testing with total test time of 19.64 Second.
shows the usage of the tester’s equipment and the equipment cost in the sequential flow. Table 3.4 has the equipment usage for the PaRent testing. According to the table, the high-cost instruments are the analog instrument used in the ADC and CC and the high power voltage current supplies. The SMPS used the high power current voltage supplies where the ADC and CC used the analog equipment. The data in Figure 3.4 shows the flow and the test time of each block. The SMPS takes 638 milliseconds and the ADC takes 658 milliseconds outside the trim test. Including the trim test, the CC takes 3396 milliseconds, the ADC takes 722 milliseconds. The utilization of the equipment is calculated as next

\[ Utilization\% = \frac{\text{Time used}}{\text{Total test time}} \times 100\% \]

Tables 3.3 and 3.4 show how the PaRent reduced the total number of tests running and increased the utilization of the high-cost testers.
Table 3.3. Tester equipment usage in sequential flow.

<table>
<thead>
<tr>
<th>Test Cost</th>
<th>Equipment Name and Cost</th>
<th>Low</th>
<th>High</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Name/Equipment</td>
<td>Digital</td>
<td>ANALOG</td>
<td>Hi</td>
<td>Power</td>
<td>Low</td>
</tr>
<tr>
<td>Open Short Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>Leakage Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>LDO</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>Power UP Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>IDDQ</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>Digital Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>SMPS</td>
<td>USED</td>
<td>NOT USED</td>
<td>USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>ADC</td>
<td>USED</td>
<td>USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>USED</td>
<td>USED</td>
<td>NOT USED</td>
<td>USED</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.4. Tester equipment usage in PaRent.

<table>
<thead>
<tr>
<th>Equipment Name and cost</th>
<th>Low</th>
<th>High</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Name/Equipment</td>
<td>Digital</td>
<td>ANALOG</td>
<td>Hi Power</td>
<td>Low Power</td>
</tr>
<tr>
<td>Open Short Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>Leakage Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>LDO</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>Power UP Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>IDDQ</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>Digital Test</td>
<td>USED</td>
<td>NOT USED</td>
<td>NOT USED</td>
<td>USED</td>
</tr>
<tr>
<td>SMPS &amp; ADC</td>
<td>USED</td>
<td>USED</td>
<td>USED</td>
<td>USED</td>
</tr>
<tr>
<td>BUCK &amp; CC</td>
<td>USED</td>
<td>USED</td>
<td>USED</td>
<td>USED</td>
</tr>
</tbody>
</table>
4.1. Introduction

Statistical process control (SPC) goes back to Western Electric’s 1956 rules in [38]. Statistical process control flags a process as being out of control based on trends of measured values (e.g., the number of data points falling outside a given number of standard deviations of the mean). The series of observations (samples) are plotted to form the patterns as shown in Figure 4.1. The patterns can be tested statistically to obtain information about the process.

Figure 4.1 has region A within 1σ from the mean value, region B between 1σ and 2σ from the mean value, and region C between 2σ and 3σ from the mean value. The rules are designed to detect any instability in the process and the rules are

1. Any data outside 3σ limits. Rule 1 line on Figure 4.1.
2. Two out of three consecutive points fall beyond the 2σ-limit (in zone C or beyond), on the same side of the mean value. Rule 2 line on Figure 4.1.
3. Four out of five consecutive points fall beyond the 1σ-limit (in zone B or beyond), on the same side of the mean value. Rule 3 line on Figure 4.1.
4. Eight consecutive points fall on the same side of the mean value (in zone A or beyond). Rule 4 on Figure 4.1.

One item that engineers in the semiconductor industry want to know is the yield of the process, given by Equation 4.1. In addition to the yield information for the process, they need to have a summary of the yield according to the lot number, wafer number, the load
Figure 4.1. Western Electrical Rules.
board number, the testers, the handler, and the probe card number. Data from the process is visualized using scatter plots, histograms, and control plots. It is much easier for the test engineer to analyze the data using charts than looking at a table of numbers. Control chart analysis is used to improve the yield and reduce the number of defects in the fabrication process which lowers the cost of the silicon.

\[
Yield = \frac{\text{Total good devices}}{\text{Total devices tested}}
\]  

(4.1)

The ATE companies used Standard Test Data Format (STDF) file to store the test information which is discussed in [39]. This is a binary format that can be converted to ASCII format which is used for yield analysis. This format is very efficient in data storage usage. However, the file can not be accessed until testing is finished for all the parts in the lot and the STDF file is closed. Therefore, any yield analysis has to wait until the test is done and the file is closed.

Data processing and yield analysis take a lot of processor time. Running them in the tester will load the tester processor and slow the testing time. The proposed algorithm for a real time monitor used Pearson’s chi-squared test method. The chi-square test requires simple calculations using the number of passing and failing units. Programming the chi-square test is easy to do without adding notable tester time.

Statistical process control observes the device parameters over time and identifies any parameters that manifest excessive variation. However, the statistical process control does not identify the root cause of the problem. It just tells when to look for possible instability in the process.

### 4.2. History of Using Data Analysis in the Semiconductor Industry

Statistical data analysis has been used in the semiconductor testing industry for many years. For example, the authors of [40] discussed how the design modeling, verification, and testing of analog ICs improved when their variations were understood. They explore the use of data mining and the re-framing of the design model order to reduce the complexity
of extracting highly nonlinear behavioral models. Furthermore, they study the use of new statistical machine learning techniques in design tools, which can speed up the design and verification process and produce more accurate results. They also cover the utilization of statistical tests to replace the specification-based tests by alternative fast and cheap measurements in the test program. Finally, the use of an adaptive test to reduce the test time by dynamically changing the limits, test flow or test content based on historical and real-time test data, is discussed.

Another method is static compaction, which is based on many highly correlated specified parameters in the chips. Static compaction obviates the need to test each parameter one by one. Using static compaction, testing a subset of the list will be enough to find any defect in the chip, and there are many systemic methods to find the right subset such as the near-optimal solution method [41], or heuristics method [42]. The heuristics method is a faster way to find a solution when the number of the specifications is large and well-formulated. The heuristics method uses a binary decision tree to find the redundant tests and eliminate them, and then compacts the complete test list into a smaller list. Having to do fewer tests will cut the test time and cut the cost of testing.

Also, using adaptive test development is to first optimizes the test list and changes the order of the test list to minimize the test time, as discussed in [43]. For mass volume production, different statistical techniques and adaptive testing are used to reduce the cost of manufacturing and testing. The adaptive test adjusts the test flow with respect to the statistical changes in the results of the tests outcomes. One of the techniques used in adaptive testing treats all the devices in the lot as identical; therefore, it is not necessary to relearn the statistical characteristics of the device with every new lot. The adaptive test algorithm schedules some of the removed tests to run dynamically on a certain location in the wafer map. However, the algorithm was modified to eliminate those scheduled tests based on the data obtained so far from running the wafer in [12].

In [44], the adaptive test is used with MST, and a device-to-device correlation with a neighboring device is used for efficient defect screenings. In [45], the information collected
during the testing of chips generated a large amount of statistical data. The data was used to find appropriate test limits to reach the goal of reducing the ratio of test over-kill to test escapes. Test escape is a defect that wasn’t detected by test program. This is could be a missing test or code mistake in the test.

Also, data mining was used to predict the post-burn-in test response. In [46], [47], and [48], a virtual probe is introduced, where a pattern and a correlation in the test data are found, and the measurement data from a small subset of units is used to predict the performance of the other units on the same wafer without taking the measurement. This will reduce the total test time of the wafer. References [49] and [50] discuss the virtual probe technique, where the test results of a small number of dies in the wafer can predict the test results on the other dies. This can be done by learning the spatial correlation of the test data across the wafer. The virtual probe in [49] was developed based on compressed sensing. The algorithm was adopted from the maximum a posteriori (MAP) estimation [51]. In addition, moving limits (where the test limits are tightened) and multiple parameter correlation techniques are used to screen the outliers in [52]. In [43], [53] and [54], different adaptive testing approaches and the moving of limits dynamically based on the measurements obtained from the neighboring dies (chips in the wafer level) are explained.

In [55], the part average analysis (PAA) tool is discussed and compared to a part average test (PAT). The PAA is a tool used by the automotive suppliers for early detection of latent defects in the components and, it is based on the PAT. The PAT is used in semiconductors to mark conspicuous dies in the wafer as possible outliers. The Automotive Electronics Council (AEC) published the first PAT guideline in 1997 in [56]. It was a guideline for using statistical techniques to establish part (chip) test limits to get good units. This will provide a known good die. The latest revision AEC-Q001 Rev-D was published in December 2011. Also, the AEC released AEC Q002 RevB which is Guidelines for Statistical Yield Analysis. It provides guidelines for using statistical techniques to detect and remove abnormal lots of ICs.

Statistical analysis has also been used extensively for outlier identification and to prevent
field returns. For example, such analysis is used to screen the outlier units from analog products during a manufacturing test in [57]. In [58], the statistical analysis of the parametric measurements in the corners of the test is used to define latent defects. In [59], customer returns are used to build a test model to prevent similar customer returns in the future. In [60], the independent component analysis method combined with a statistical post-processing technique to remove the spatial variation of the supply current in the quiescent state (IDDQ) is discussed. This method can remove the outlier dies that pass the production test program but are not within the normal distribution of the data.

All of the previous sophisticated methods are post-processing methods and none of them is a real-time monitor as proposed here. Moreover, some of the methods work on the wafer level only and not the packaged units. On the other hand, the proposed algorithm works on both wafer level and packaged units.

4.3. The Reliability of the Fit Test and the Hypothesis Test

The reliability of fit test is a statistical test to measure how well the observed data fits the given statistical model, as discussed in [61], [62], and [63]. Those measurements are used in statistical hypothesis testing to find whether the samples are drawn from identical distribution using the Kolmogorov-Smirnov test or whether the outcome frequencies follow a specific distribution as with Pearson’s Chi-square test.

A hypothesis of a statistical relationship such as equality of distribution existing between the two data sets is tested against the null hypothesis that proposes no statistical relationship between the two sets of data. The test of the hypothesis determines if the null hypothesis will be rejected for a pre-specified threshold probability. The pre-specified threshold is defined as the level of significance. Statisticians recommend reporting both the p-value and the appropriate confidence limits with the correct interpretation.

4.4. Pearson’s Chi-squared test

Pearson’s Chi-squared test ($\chi^2$) is used to evaluate three types: homogeneity, goodness of
fit, and independence. More details are discussed in appendix A.2. χ² Test for Independence is used in the site yield monitoring due to its simplicity and speed in the calculation. Therefore, the addition in the test time will be minimal, and the result of the test will indicate the possibility of an issue in any of the sites. Moreover, data from the test results get saved to a special format in the database where it is inaccessible until all the material has been tested. This prohibits using more advanced or complicated methods in the tester to make sure that all of the time is spent on the testing only to keep cost low.

Figure 4.2 shows the different variables that could affect the yield, starting with selecting the fabrication process, the facility, and the design process. Those variables are controlled in the early stage of the design. The second step is the assembly stage and the testing stage. Testing has many variables that could be affected by the continuous usage in the production environment. Those variables are related to the many boards, testers, handlers, probers, and sockets.

Any decay in any of the equipment in the test stage will affect the yield. However, the yield result is designed to be independent of the tester, site, prober, the handler, or the socket used in testing. Therefore, the χ² test of independence can be used to flag any possible issue in the yield due to any failure in the test system set. The χ² test will indicate the possibility of an issue, but it will not identify the root cause of the failure.

4.4.1. Limitation of Chi-square Test

Paper [64] discusses the limitation of the chi-square test, starting with the fact that the χ² test can be applied to qualitative data classified into categories and not to continuous data. It can be shown mathematically that χ² approximation is not valid if the expected value of any frequency in the contingency table is less than 5. In other words, the χ² test is an asymptotic test; therefore, the sample size must be large enough for the χ² approximation to be accurate. If an expected value is less than 5, Fisher’s exact test is recommended.

Yates correction is a continuity correction, which is used because the χ² distribution is continuous, yet the data in the table are dichotomous. As a result, the χ² values will
be biased upwards. The Yates correction should always be employed, although its effect diminishes as the sample size increases. This implies a large sample size. However, with a small size sample, Yates correction is necessary to obtain valid p-value and confidence intervals.

Another limitation of the $\chi^2$ test is that the p-value is dependent on the sample size. The larger the sample size, the smaller the p-value will be, even if the overall effect size is constant. For the coin toss experiment, a doubling of the sample size will result in a p-value that is four times as small as the one calculated even if the relative proportions of heads and tails remain the same. Thus, it is easy to “fool” a $\chi^2$ test with a large enough sample size. Appendix A.1 discuss the two kinds of errors in hypothesis testing.

4.5. Variation in Measurement

Any measurement has a degree of uncertainty due to different sources. Reporting any
measured value requires an estimation of the level of confidence. This allows the reviewer of the reported results to make a judgment about the experiment and its quality. Also, it enables the comparison of data to a similar experiment or theoretical prediction. Deciding the fidelity of the data depends on specified uncertainty or variation of the measurement. Making a measurement is finding the ideal quantity using the best available resource within our capability at that time. Measuring the same quantity a few times using the same method or different methods might result in slightly different values. Therefore, the measured value is usually reported as

\[ Measurement = \text{best value} \pm \text{uncertainty}. \]  \hspace{1cm} (4.2)

The uncertainty in the equation could be due to a systemic error or a random error. Random error is due to the limitation in the instrument used in the measurement. The random error can be reduced by using a more accurate instrument or averaging a large number of observations.

Systemic error is a consistent inaccuracy in the same direction. That error can be reduced by calibrating the instrument against a standard and applying a correction factor to compensate for the effect of the bias in the measurement. Averaging a large number will not reduce the systemic error.

Figures 4.3, and 4.4 show Normal data distributions of data with the same mean, and different variances. Figure 4.3 has data without any systemic error. However, Figure 4.4 has negative systemic errors.

There are many sources of error when taking measurements. Those sources are due to the instruments, the users, and the setups. All of the above will cause a variation in the results. With the semiconductor industry testing, there are more sources of variation in the design process, the fabrication process, probing process, packing process, and final testing process.

The instrument’s precision and accuracy with any other factor that might affect the experiment will limit the uncertainty of any single measurement.
Figure 4.3. Normal distribution with mean=0 and different variance 1, 2 and 3.
Figure 4.4. Normal distribution with different variance and mean=-0.5 which represent negative systemic error.
4.6. Variation in Semiconductor Industry

Each step in the semiconductor manufacturing process contributes to variation of the final product. The process variation in the fabrication will cause each wafer to vary from one another. Moreover, each die in the wafer will be different from the others. However, the process could shift over time. This shift could cause changes in the yield. Therefore, the deviation in the yield has to be observed to find and adjust any variation in the process. Moreover, in the high volume production environment, the need to use different machines such as testers, prober, handler, load boards, introduces more variation. Likewise, in the multi-site solution, each site’s test result will vary from the others.

Releasing chips into production requires sending many load boards to accommodate the production volume. Usually, load boards have exact design and similar components. However, the results of the same site number could vary between different load boards. Moreover, there is no guarantee that variations will not exist between the different testers even when using the same chips on the same sites and the same board.

Site-to-site variation may also exist in MST due to variations originating from discrepancies in the version of the board layout, tester resource arrangements, and handler/prober alignment. For example, in the case of analog test, the analog waveform generator, voltmeter, DC sources, and other sources/meters come to the test head in a specific order. With a high number of sites, it is unlikely to have an identical resource order for all sites. Moreover, the sites will have different trace length due to their location on the board. In an automated test, the handler/prober pushes the devices/dies to the prober/socket.

In MST, the pressure on all sites will not be equal. The resulting measurements may be affected due to the impedance mismatch in socket/prober.

As the number of sites increases, the number of external components on the board increases. Therefore, the probability of component degradation and faulty components increases as the number of sites increases. A defective component may impact the total yield of all sites on the board and the yield of the site with that component. The variation of measured data from this site would be idiosyncratic from that obtained from the other sites.
and the other boards. As the distribution of the measurements shifts for that site, some bad units might pass the limits and good units might fail the limits as shown in the figure 4.5 below. The higher the variation in the test yield among sites, the less confidence there will be in the test results.

Figure 4.5.  Faulty component causes a shift of the data by one standard deviation.

In spite of the possible increase in the test measurement variation as the number of sites increases, the demand is to have more sites in production due to the obligation to fulfill the high-volume for tested parts at a low-cost. Reference [22] shows that multi-site solution is one of the most effective approaches to reduce the cost of testing. Thus, it is common to find high-pin-count System-on-Chip (SOC) devices running on quad sites, octal sites, and even higher counts of sites in a production environment.

There is a lot of effort and time spent to minimize the variation before releasing the
product to the production environment. However, the production environment and the continuous usage of the different equipment and boards could induce shifts in variation to shift which leads to changes in the yield. Therefore, it is critical to have a real-time monitor in production to sustain the high quality of the tested chips.

4.7. Minimizing Test Variation in Semiconductor Industry

Test measurement variations are common anxieties in semiconductor testing. The design and manufacturing stage is a part of the variation process in the digital and analog design. Test measurements have other sources of variation, especially for analog tests. For example, testing sensitivity due to program vulnerability could cause variations in the test results, where testing one unit many times using the same test program gives different results. Usually, this vulnerability is due to the short wait time in the test program before measuring the output of the chip, the inadequate number of samples in the measurement, or a glitch due to programming oversight that could cause the device to be in the wrong mode or that resets the device.

Tester-to-tester and load-board to load-board variations add to the overall test variations. The final test and probe boards could also inject additional variations due to the design of the boards, the number, and type of components used in the boards, the number of sites, the use of sockets, and the length of time the boards have been in use, which may lead to mechanical and electrical stress. The mechanical damage to the probe needles and the wafers, its impact on the yield, and suggested solutions are all discussed in [65], [66], [67], and [68]. Site-to-site variations and within-site variations that arise due to the wide variety of load board layouts, alignment, and component issues listed earlier are common. Tester-to-tester and load-board to load-board variations add to the overall test variations.

Process capability ($C_p$) compares process output to the specification limits of a product parameter. $C_p$ is the consistency and repeatability of a manufacturing process relative to the customer’s specification limits. The next equations give the definition for the $C_p$ depending on the specification limits.
\[ C_p = \frac{USL - LSL}{6\sigma} \] (4.3)

\[ C_p = \frac{USL - \mu}{6\sigma} \text{ if no LSL in the specification} \] (4.4)

\[ C_p = \frac{\mu - LSL}{6\sigma} \text{ if no USL in the specification} \] (4.5)

The process capability index \( C_{pk} \) is defined in Equation 4.6, and it is a statistical measurement of the process capability to produce output within the specification limits.

\[ c_{pk} = \text{minimum}\left[\frac{\mu - LSL}{3\sigma}, \frac{USL - \mu}{3\sigma}\right] \] (4.6)

where: \( \mu \) is the mean of the data, USL and LSL are the upper specification limit and the lower specification limit, and \( \sigma \) is the overall standard deviation. The standard deviation is due to reproducibility errors and repeatability errors in the system.

\[ \sigma = \sqrt{\sigma_r^2 + \sigma_R^2} \] (4.7)

Where: \( \sigma_r \) is the standard deviation of the repeatability data; \( \sigma_R \) is the standard deviation of the reproducibility data. Note that repeatability refers to testing the same unit on the same setup multiple times and obtaining repeatable data. Reproducibility refers to testing the unit on different setups (sites, boards, or testers) and being able to reproduce the same data.

Gauge repeatability and reproducibility (GRR) is one of the methods used to study the variance of the test systems in the semiconductor industry, and GRR is defined in [51] as

\[ GRR = \frac{100}{C_p} \] (4.8)

The lower the GRR value, the higher the \( C_p \) is. So, the variation is less, and the process is more repeatable. Moreover, the data is centered in the middle of the two limits.

Multiple papers (e.g., [9], [69], [70], [71], and [72]) explain the process of finding GRR.
and the different equations used. GRR is used in many different industries, and it started with the Automotive Industry Action Group.

In the semiconductor industry, two load boards, two testers, and a number of devices under test (DUT) are used to calculate GRR. The number of DUTs in the GRR should be at least four times the number of sites on the boards for the system to be reproducible. The GRR is done by writing serial numbers on each unit and then testing every site at least five times using two testers and two boards. The units serializing is critical in the calculation of the test repeatability standard deviation of each unit in the different sites, boards and testers.

The number of DUTs should be four times the number of sites; thus, for a quad site solution, you need 4x4=16 DUTs. Testing each of the DUTs five times would result in 16x5=80 setups to perform. Running the test program using two boards and two testers is necessary to calculate the reproducibility standard. Therefore 80 x 2 x 2= 320 setups are needed to get the reproducibility and repeatability standard deviation for a quad site solution. It is essential to keep track of the DUTs, boards, and testers in calculating the reproducibility, repeatability, and total standard deviation, which are used in the \( C_p \) and GRR calculations.

The lower the GRR value is, the more repeatable and reproducible the test is. According to the Automotive Industry Action Group (AIAG), the GRR should be less than 10% to have a system with acceptable repeatability and reproducibility [69]. This also means that the measured values do not depend on the site, the tester, or the board.

Obtaining a GRR within limits may require enhancement in the test program. With a test program with few hundred tests, it takes time to collect the data and process it to find, modify and fix any test with a high GRR value. Investigation, modification and recollecting the data could take weeks in some cases. It could also increase test time due to the change in wait time in the program to have a solid signal from the chip or the tester. Moreover, the number of measurement samples might increase to get a more precise measurement and enhance GRR values. In extreme cases, the accuracy of the tester is less than the
specification, and the GRR values are higher than 10%. This will lead to renegotiating the specification’s limits.

In spite of all the hard work and long hours, releasing the program and hardware to mass production might run into issues due to concerns in the production site testers, handlers, and other environmental conditions. It is common to run a few material lots before going to full production. However, this will not eliminate the need for a real-time monitor.

4.8. Yield Monitoring

The GRR calculation is done before releasing the board, the program, and the device to production and in the early stages of production. However, after some time in production, some of the components on the load boards might become faulty or damaged. This will affect the test results and reduce the quality of the testing, where the data will have outliers, level shifts, and variance changes. The customer might get outlier devices or even bad units if the changes in the test system are not addressed. Therefore, it is essential to have a real-time monitor to detect these issues as early as possible.

Figure 4.5 shows that a shift of the data by one standard deviation will cause a fall out of 2.27% from the original fall out of 0.27%. Moreover, 0.13% of the failing unit will pass the limits. In the other words, bad units will be shipped as good ones. They red triangular shows those units.

If one site or load board becomes faulty, the reliability and the yield of the measurements taken from the concerned site or board will be affected. Most likely, it will be less than the other boards and sites. This presents both an opportunity and a challenge. If we can identify the inconsistent quality of the responses coming from that unique site or board, we can not only define the sources of the problem, but also fix them. The challenge prevails in finding a low-cost, effective way to distinguish such an obstacle in real time.

Today, post-processing the yield data is the most popular way to watch variations in the production data across sites, wafers, etc. For example, post-processing is a common way to identify the outliers in probe testing, where a wafer map of the test results is saved in the
database. Thus, the outliers can be found easily by data mining, and the suspicious units can be marked on the wafer as discussed in [58]. However, it is absurd to use post-processing in the final test since it does not have a map to track the packaged unit’s location. Most of the time, if a problem in the yield or test setup of the packaged units is suspected, all of the tested units have to be retested.

Additional obstacles occur in the case of One-Time-Programmable (OTP) memory devices. Modern SoC designs use such memories for multiple functions. For example, the OTP may be programmed to hold an encryption key for security applications. Boot code may also be stored in the memory to configure the device’s levels, startup sequence, and timing of different blocks. In the case of trimming analog circuits, a code to compensate for the deviation from the target specification in the analog circuits and sensors due to the variation in the chip process and packing operations could be found and programmed into the memory. However, if the board or site on which the device is tested is defective, then the configuration parameters programmed into the OTP memory may be inaccurate. This requires the affected units to be scrapped. Scrapping all of the good units will affect the delivery to the customer. It could cause fines and extra cost for the chip manufacturer. It is essential to have a real-time monitor in production that flags any possible issue in yield as soon as possible.

4.9. Yield Data of 8 site solution

Figure 4.6 shows a lower yield than the expected value for the output resistor measurement. On the other hand, Figure 4.8 shows the measured output resistor value ($R_{out} (\text{ohm})$) for a lot within the expected yield value. The test has a lower limit of zero and a higher limit of two ohms. The test passes if the measured value is between the limits and fails if it is outside the limits. Figures 4.7, and 4.9 show the histogram for each site in both lots.

Table 4.1 shows the tally of the pass and the fail results by the site used to test the reference current ($I_{ref}$) of the chip. It shows the result of two sets of units. Each set has the results of 240 units. The unit passes if the measured current is between 7uA and 13uA and
fails if the measurement is outside the limits. Figures 4.10 and 4.11 show the measurements of Iref and the histogram of each site of a lot with 5324 units. Both figures show that site one has many outliers. The test results and the lower yield will be discussed more in the next chapter. The proposed procedure in the next chapter will process the data in the table using the $\chi^2$ test to flag the possibility of a yield issue.

![Figure 4.6. Rout(Ohm) test result with the low yield. High percentage of units failed the 2 ohm upper limit.](image)

Table 4.1. T1 observed data for 480 units.

<table>
<thead>
<tr>
<th>Site</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed Value(1st set)</td>
<td>Fail</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Pass</td>
<td>25</td>
<td>36</td>
<td>34</td>
<td>34</td>
<td>25</td>
<td>24</td>
<td>26</td>
</tr>
<tr>
<td>Observed Value(2nd set)</td>
<td>Fail</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Pass</td>
<td>20</td>
<td>28</td>
<td>28</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>
Figure 4.7. Rout(Ohm) test histogram with the low yield. All of the sites have high yield loss.

Figure 4.8. Rout(Ohm) test result of a lot with the expected yield. Only a small percentage of units failed the 2 ohm upper limit.
Figure 4.9. Rout(Ohm) test histogram of a lot with the expected yield. Only small percentage of the units failed the 2 ohm high limit in this lot.
Figure 4.10. T3 (Reference Current Test) results by site for the lot with issue. One site is failing the test and the other sites have no issues.
Figure 4.11. T3 (Reference Current Test) histogram data by site for the lot with issue. All the sites have the data centered around 10.6µA, but site one has many units around 20µA.
5.1. Introduction

Running production requires delivery of high-quality units on time. The hurdle is to monitor continually the quality of the units and the test setup. It is not easy to assess the quality of the tests as soon as the units start to fail. Statistical methods help in determining the quality of the test yield and setup. Moreover, statistical methods can be used to calculate the sample size needed to make a well-informed decision about test quality.

Statistically speaking, a small sample size may not result in reliable decisions. On the other hand, the larger the sample size, the more accurate decisions are if the data is accurately collected. However, sample sizes that are too large consume a lot of time and resources. The required sample size depends on the level of accuracy desired, the probability of detecting a correct result, and the probability of making a type one error.

The algorithm proposed here discerns the yield of each site on the board using the $\chi^2$ test. The goal is to quickly identify and flag any site in a multi-site setup as soon as it begins to yield less than the expected yield from the design process and from the other sites running on other testers using different wafers or chips.

5.2. Real Time Yield Monitoring Algorithm

Figure 5.1 shows the proposed algorithm to monitor the yield. The algorithm keeps track of the total number of units tested, the number of passing units in each site, the number of failing units in each site, and the number of generated alarms.

The algorithm demands a minimum number of tested units to have enough samples to check the site yield compared to the predefined expected yield and check if the yield depends
on the site. The dependency check results will flag sites where the observed yield of any site in the window is less than expected. The window size defined in the proposed algorithm is defined as the minimum number of tested units to run the $\chi^2$ test. The window size for the eight-site solution is two hundred forty units. The number of failing and passing units at each site is needed to calculate the yield of each site. If the number of passing units is less than the expected yield value, the $\chi^2$ test will run. However, if it is not less than the expected yield the $\chi^2$ test will not run. This will save test time if test time is especially critical. The “RunDependency” variable in Figure 5.1 will be false.

Figure 5.1. Real-time yield monitoring algorithm.
In an eight-site solution, there are seven degrees of freedom. This is because the eight sites constitute 8 categories (i.e. columns) in the contingency table, and there are two possible test results: pass and fail (corresponding to the rows of the table). According to Equation A.1, the corresponding degrees of freedom equals (8-1) (2-1) = 7. The critical value for the $\chi^2$ with 7 degrees of freedom is 14.07 if the significance level is 0.05 and 18.48 if the significance level is 0.01.

The critical value will be used to reject or fail to reject the null hypothesis, where the null hypothesis is that the test results are independent of the site used to test the part. In the algorithm shown in Figure 5.1, once an appropriate number of parts have been tested to fill the desired window, a dependency check function will be called. Figure 5.2 has the pseudo-code for the function. If any site has more failing units than the expected yield, then the pseudo-code in the function will be executed.

Figure 5.2 shows the pseudo-code of the function to examine the dependency using the $\chi^2$ test. The function inputs are the anticipated yield ratio, the number of passing and failing units in each site, and the critical $\chi^2$ value. The function calculates the $\chi^2$ value of the passing and the failing units and returns true if it is greater than the provided critical $\chi^2$ value. Also, the function calculates the residual of all the sites and flags the site with the highest residual, distinguishing the site with a potential issue.

As shown in Figure 5.1, if there is evidence that the sites are dependent, the algorithm generates an informational alert to be sent to the list of the operators. The alarm will involve the calculated residual of each site, the site with the highest residual value, and the name of the test with the potential issue. The residual information is essential to determine the site with a possible issue. In the case of a second alarm, the operators will be asked to take action. The action depends on the device, the production engineer, the history of the device in production, and the customer’s demands.

The same site with two alarms will indicate that the site likely has an issue. The dual alarm requirement, before action is taken, is utilized to decrease the probability of false alarms suspending production. Also, it allows for more data collection before suspending
function Check_Dependency_Test(Passedunits, Failedunits, 
    CHI_SQUARE_CRITICAL_VALUE, 
    ExpectedYield, var max_Res): bool 
{
    CHI_SUM = 0
    for (i = 1 to NumbSites)
    {
        if ((Passedunits[i]+Failedunits[i])>0)
        {
            Y[1,i] = Passedunits[i];
            Y[2,i] = Failedunits[i];
            E[1,i] = (Y[1,i]+Y[2,i])\*ExpectedYield
            E[2,i] = (Y[1,i]+Y[2,i])\*(1-ExpectedYield)
            RESIDUAL [1,i] = (Y[1,i]-E[1,i])^2/E[1,i]
            RESIDUAL [2,i] = (Y[2,i]-E[2,i])^2/E[2,i]
            CHI_SUM = CHI_SUM+RESIDUAL [1,i]+RESIDUAL [2,i]
        }
    }
    max_Res = find_max_location(RESIDUAL)
    if (CHI_SUM > CHI_SQUARE_CRITICAL_VALUE)
        return true
    else
        return false
}
the test. The extra data collected will help illuminate whether the problem is affecting a single site or multiple sites. In the case that the data shows that an alarm is arising from a different site, this indicates one of the following:

1. Board issue due to multiple sites’ issues.
2. Handler issue.
3. Probe issue.
4. Tester issue.
5. In the final test due to dicing the wafer, or packaging process.
6. In the probe test due to a process issue.

In any case, more investigation is needed to determine the root of the issue. The complexity, and the interdependency of manufacturing semiconductor chips makes isolating the source of the problem very complicated.

5.3. CASE STUDY

The algorithm was implemented using R, an open-source programming language for statistical computing and graphics [73]. The algorithm was verified using data from four different lots. The expected yield by the designer and the characterization was around 97%. The initial yield of many tested lots was approximately 97%. The goal was to discover whether the algorithm could have identified the presence of the problem more quickly during production, had it been applied at that time.

The production program had 378 tests, and each lot had 5432 units. The four lots studied involved two lots with the expected yields and two lots with lower than expected yields. The yield of the third lot was 48.8%. T1 and T2 had the most loss in that lot.

T1 represents the resistor of the block inside the chip. The upper limit for the resistor is 2 ohm. Figure 5.3 shows the measured resistor value for the four lots. It has the results
Figure 5.3. T1 (Output Resistor) Results for all lots.

The resistor value inside the chip is calculated by finding the amplification ratio between the output voltage and the input voltage. The input voltage is applied by forcing a current through a resistor on the board. Each site has a similar resistor value on the board. Thus, any change in the resistor value will change the input voltage to the chip and the output voltage of the chip will change too.

Figure 5.5 has the dependency results for T1 in lot 3 using a significance level of 0.01. The plot shows the dependency analysis results for each window (240 units tested). Each lot has 5534 units in total tested using 8 sites. The algorithm will run the $\chi^2$ test in 240 units which means that 23 windows have 240 units, and the last window will have 14 units. Window 24 will have not have enough data to run the test. The algorithm is written where the default is that the all of the sites’ test results are independent of the site used to test the parts. This is the reason for window 24 to be independent.

Figure 5.6 plot shows the site with the highest residual for the window when the test
Figure 5.4. T1 (Output Resistor) Results and histogram for the lot with lower yield.
results are dependent on the site. Clearly, multiple sites have issues. Figures 5.7 and 5.8 have the results of dependency analysis using a limit of 0.05 for the p-value.

In Table 5.1, the residuals of all sites in the first window are shown for test T1 and T2 of lot 3 and test T3 of lot 4. The residual is in red if it is larger than the critical $\chi^2$ level.

According to Figures 5.5, 5.6, 5.7, and 5.8, there are issues in different sites and different windows. However, the residual results for test T1 in Table 5.1 show that all sites have issues since the $\chi^2$ value is greater than the $\chi^2$ critical level. The overall loss was around 52% which is more than the expected loss of 3%.

Table 5.1. The calculated residual for the first window with an issue.

<table>
<thead>
<tr>
<th></th>
<th>Site</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Fail</td>
<td>483.84</td>
<td>284.86</td>
<td>534.81</td>
<td>350.55</td>
<td>271.06</td>
<td>282.67</td>
<td>342.00</td>
<td>206.12</td>
</tr>
<tr>
<td></td>
<td>Pass</td>
<td>14.96</td>
<td>8.81</td>
<td>16.54</td>
<td>10.84</td>
<td>8.38</td>
<td>8.74</td>
<td>10.58</td>
<td>6.37</td>
</tr>
<tr>
<td>T2</td>
<td>Fail</td>
<td>0.99</td>
<td>79.06</td>
<td>1.08</td>
<td>0.90</td>
<td>0.39</td>
<td>0.90</td>
<td>0.84</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>Pass</td>
<td>0.03</td>
<td>2.45</td>
<td>0.03</td>
<td>0.03</td>
<td>0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>T3</td>
<td>Fail</td>
<td>95.81</td>
<td>0.78</td>
<td>0.84</td>
<td>0.75</td>
<td>1.05</td>
<td>0.99</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>Pass</td>
<td>2.96</td>
<td>0.02</td>
<td>0.03</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Test T1 had an issue on all sites, indicating a potential load board, tester, or lot issue. However, debugging the issue eliminated any issue on the lot, the tester, or the probe. The issue was the load board. Each site has an analog circuit on the board. The output of the onboard circuit gets amplified inside the chip. Small variations will lead to big output changes. The debug confirmed that the problem was the resistor tolerance used in that circuit. The circuit used 0.1% tolerance resistors to generate the input to the chip. However, that tolerance was not enough. Two actions were needed to fix the issue. First, a resistor with 0.01% tolerance has to be used. Secondly, an open socket calibration and site-based correlation factor have to be programmed.

The result of T2 for all sites is in Figure 5.9. It is clear that T2 has a defect in the units
Figure 5.5. Dependency Check Function results for T1 using significance level of 0.01. All of the windows show the test results depend on the site used in testing. The exception is window 24 because window 24 has 14 units which less than 240. Thus, the $\chi^2$ is not run and it assumed to be independent.
Figure 5.6. Site location for Dependency Check Function results for T1 using significance value of 0.01. All of the windows show the test results depend on the site used in testing due to an issue on the load board affecting all the sites.
Figure 5.7. Dependency Check Function results for T1 using significance value of 0.05. Window 24 has 14 units which less than 240. The $\chi^2$ is not run, and it assumed to be independent.
Figure 5.8. Site location for Dependency Check Function results for T1 using significance value of 0.05. All the sites have an issue in the test result due to an issue on the load board. The location of the largest residual is not same. The problem was on all of the sites due to load board issue.
Figure 5.9. T2 (Functional Test) results for all lots.

Figure 5.10. T2 (Functional Test) results for a good lot.
or the measurements between unit number 11,000 and 16,000, which represent the lot with a possible issue. The data for those points are in Figure 5.11, while, Figure 5.10 has the measurement for a lot without the defect.

Figure 5.12 has the results for T2 in lot 3 using a significance level of 0.01. The plot presents the dependency analysis results for each site in the window. Figure 5.13 displays the site with the largest residual for the window when the test results are dependent on the site. It is clear that windows 20 and 23 have a dependency issue. Figure 5.14 and 5.15 shows the results of dependency analysis using a significance level of 0.05. Using a larger significance level flags a potential problem earlier than when a significance level of 0.01 is used. It exposes the issue in the first and second window. Testing only four hundred and eighty units flags the defect using the algorithm with a significance level of 0.05.

In Table 5.1, the residual of site 2 at window twenty has the largest value. It is more than the limit of 18.84 by many factors. However, the dependency analysis results in the first two windows showed a $\chi^2$ value less than 18.84, but greater than 14.07. So, they are not shown in the table.
Figure 5.12. Dependency Check Function results for T2 of Lot 3 using significance value of 0.01. Windows 20 and 23 shows that the test depends on the site used to test it.
Figure 5.13. Site location for Dependency Check Function results for T2 of Lot 3 using significance value of 0.01 using $\chi^2$ Test. Windows 20 and 23 shows that Site 2 has an issue. It has a higher yield loss than the other sites.
Test 2

Significance level $\alpha = 0.05$ using $\chi^2$ Test

Figure 5.14. Dependency Check Function results for T2 using significance value of 0.05. Windows 1, 2, 20, 22, and 23 shows that the test depends on the site used to test it.
Figure 5.15. Site location for Dependency Check Function results for T2 using significance value of 0.05. Windows 1, 2, 20, 22, and 23 shows that Site 2 has an issue.
Debugging the problem revealed that the wait time was the cause for the failures. Increasing wait time between making the measurement and setting the chip in the right mode and setting the input signal to the right level in the program for that test fixed the problem. T1 issue was masking the issue of T2 since T2 runs after T1 passes. However, T2 does not run if T1 does not pass. The execution of the test flow for any site will stop for that site when it fails any test. The yield of T2 would be worse if the test flow did not stop at T1 failures.

![Ref Current of all lots](image)

Figure 5.16. T3 (Reference Current Test) results for all lots.

The fourth lot’s yield was 91.39% and T3 was the source of all failures in site 1. Figure 5.16 has the data for all of the lots. It is clear that one lot has more failures than the other lots. The data for the good lot is in Figure 5.17 and for the lot with an issue is in Figure 5.18. T3 represents the measurement of the internal reference current in the chip. The expected measurement is 10.6uA; therefore, the instrument is set to clamp on 20uA which means that the site with the issue is showing 20uA.

The yield for site 1 was 57.76% for this lot. Figures 5.19, 5.20, 5.21, and 5.22, show that
Figure 5.17. T3 (Reference Current Test) results for good lot.

Figure 5.18. T3 (Reference Current Test) results for a lot with issue.
Figure 5.19. Dependency Check Function results for T3 using significance value of 0.01. All of the windows show the test results depend on the site used in testing, but window 24 because window 24 has 14 units only which less than 240. The $\chi^2$ is not run and it assumed to be independent.
Figure 5.20. Site location for Dependency Check Function results for T3 using significance value of 0.01. Site 1 has an issue that caused the yield to be less than the expected yield. Window 24 does not show any site since it does not have enough samples to run $\chi^2$ test.
Figure 5.21. Dependency Check Function results for T3 using significance value of 0.05. All of the windows show the test results depend on the site used in testing, but window 24 because window 24 has 14 units only which less than 240. The $\chi^2$ is not run and it assumed to be independent.
Figure 5.22. Site location for Dependency Check Function results for T3 using significance value of 0.05. Site 1 has an issue that caused the yield to be less than the expected yield. Window 24 does not show any site since it does not have enough samples to run $\chi^2$ test.
site 1 is a concern with a significance level of 0.01 and 0.05. Also, Table 1 shows that the \( \chi^2 \) value for that site is 95.81. It is more than the limit of 18.84 by many factors. The same issue surfaces in all of the windows.

Debugging the issue revealed that the tester’s instrument in site 1 was out of calibration. Calibrating the tester fixed the problem.

Using the proposed algorithm allows the problem to be flagged earlier than the occurrence with a post-processing analysis based on the yield. For example, it would take two windows and a total of 480 tested parts to recognize the issue with our real-time monitoring algorithm for T1 as shown Figure in Figure 5.6, T2 as shown in Figure 5.15, and T3 in Figure 5.20. For example, finding the problem after two windows corresponded to 249 failing units in T1. This is true when \( \alpha = 0.05 \), but it took longer time in T2 for \( \alpha = 0.01 \). This is does not means that \( \alpha = 0.05 \) is better than \( \alpha = 0.01 \), but it means that \( \alpha \) value should be selected carefully. This selection process will depend on the cost of the chip compare to the cost of pausing the testing.

The post-processing requires the completion of testing of the whole lot to analyze the yield and identify any problem if it happens. This will produce 2699 units failing in T1 test. In the other hand, T3 will produce 256 failing units. Some of those units have to be re-tested to debug the issue and find the root source of the problem. As we saw, the problem was not the unit, but the problem was the load board, program, and tester’s issue. After this, all of the failing units must be re-tested to recoup the losses. This is an additional cost that will affect the bottom line of profit. Post-processing will not stop the usage of the tester, the load board, and the program until a resolution is reached about the source of the concern. So, this will introduce more wasted time and wasted units. Moreover, re-testing units will require getting the failed units or wafer and setting them with the board on the tester.

In contrast, using the proposed algorithm would flag the site issue with only 27 failing units in the first 2 windows, not 256 failing units in the lot. This way the test flow will stop testing after testing four hundred units with only twenty-seven failures. The size of failing units using the algorithm is one-tenth the size of failing units using the post-processing.
Finally, the analysis of the other two lots using the proposed algorithms did not show any concerns. The analysis used significance levels of 0.01 and 0.05. This confirmed that the proposed algorithm flags the issues as they emerge. Moreover, it did not produce any false alarms in the data we used.

5.4. Real-Time Monitor Pros/Cons

The proposed approach may also allow problems to be found that would otherwise remain invisible in relatively high yields. For example, in the case of T2, with only fifty total failures in the lot, the failures might have been overlooked because the total loss of the lot is less than 1%. However, with the proposed windowing algorithm the problem would be flagged and alarms would be sent. Also, the failing site would have been automatically identified. Failing to identify a test issue may lead to poor test quality. This could lead not only to yield loss but also to less trustworthy test results overall. Note that when good parts fail and are scrapped, the customer delivery commitments may be jeopardized, and the cost may increase due to obligations for testing more lots with more resources (e.g. testers, load boards, and operators).

Once a concern is flagged, several responses are possible. For example, flagging a problem and identifying the site in question will enable the operators to disable that site. This will run the material without causing any possible damage to the units that would otherwise be tested at that site. The wafer, the probe card or the load board in the final test could be damaged if testing is not stopped. Alternatively, the operators may be able to switch the load board to a backup board and continue testing to meet the commitment to customers while debugging the board with the issue. Another possibility is to switch to the diagnostic program for the test board and the tester to better identify the problem. In any case, flagging the issue early is essential to give a timely warning to the test engineer and the management to resolve any possible issue with any customer commitment addressed.

However, the algorithm will increase the test time for the program. The extra test time depends on the number of sites, and/or number of tests. The algorithm adds time to every
test to calculate the number of passing and failing units, the total number of tested units and the yield per site. The time is measured using the number of cycles as shown in the next equation.

\[
\text{Number of Cycles} = \text{Number of Tests} \times (2 + 4 \times \text{Number of Sites}) \tag{5.1}
\]

Also, when the total number of tested units reaches the size of the window, the algorithm starts new calculations. These calculations add a number of cycles as in the next equation

\[
\text{Number of Cycles} = \text{Number of Tests} \times (10 + 8 \times \text{Number of Sites}) \tag{5.2}
\]

The equation calculates the number of cycles needed to run the $\chi^2$ test. The speed of the tester’s processor will decide the run time for the code. However, this time will not affect the test if the yield result is not less the expected yield.

Each block in the chip will have a list of tests. Instead of running the $\chi^2$ on the individual test, the $\chi^2$ can run each list as one set. This will reduce the number of cycles added to the original test time.

### 5.5. Real-Time Monitor Sensitivity

Using $\chi^2$ test result to find if the yield result is independent of the site used in testing the unit will be quick and easy to add to the test program. Nonetheless, it will be influenced by the expected yield, the number of samples used per site and the total number of sites used. Therefore, it is important to determine the appropriate sample size to maximize true positive and minimize the false positive. The larger the sample size is, the more likely false positive will accrue. However, this will require more time to collect the samples and the test time is an expensive option to add.

Table 5.2 has the yield result of a test program with N sites. Each site has n samples and the number of passing unit for the site i is $O_i$. So, the number of failing units for that site is $n - O_i$.  

85
Table 5.2. Contingency Table for the yield of N sites

<table>
<thead>
<tr>
<th></th>
<th>Site1</th>
<th>Site2</th>
<th>...</th>
<th>SiteN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>$O_1$</td>
<td>$O_2$</td>
<td>...</td>
<td>$O_n$</td>
</tr>
<tr>
<td>Fail</td>
<td>$n - O_1$</td>
<td>$n - O_2$</td>
<td>...</td>
<td>$n - O_n$</td>
</tr>
<tr>
<td>Total</td>
<td>n</td>
<td>n</td>
<td>...</td>
<td>n</td>
</tr>
</tbody>
</table>

Table 5.3 is the contingency table assuming that one site did not have the expected yield, but the others did. The table can be used to calculate the sensitivity of the $\chi^2$ test due to the yield of that site.

The assumption that only one site does not meet the yield expectation will simplify the calculation of the sensitivity of $\chi^2$ due to change in the real yield, the number of sites and the specified critical value. The equations 5.3 - 5.8 show the $\chi^2$ calculations. Equation 5.7 shows the relation between the number of passing units $O_i$ in the samples and the expected number of failing units and the failing units and the critical value $\chi^2$.

Table 5.3. Contingency table when site i had different yield than the expected yield.

<table>
<thead>
<tr>
<th></th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Observed</td>
<td>Expected</td>
</tr>
<tr>
<td>Site 1</td>
<td>$O_1$</td>
<td>$O_1$</td>
</tr>
<tr>
<td>Site 2</td>
<td>$O_2$</td>
<td>$O_2$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Site i</td>
<td>$O_i$</td>
<td>$E_i$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Site N</td>
<td>$O_n$</td>
<td>$O_n$</td>
</tr>
</tbody>
</table>
\[
\chi^2 = \frac{(O_i - E_i)^2}{E_i} + \frac{(n - O_i - (n - E_i))^2}{n - E_i}
\] (5.3)

\[
\chi^2 = \frac{(O_i - E_i)^2}{E_i} + \frac{(n - O_i - n + E_i))^2}{n - E_i}
\] (5.4)

\[
\chi^2 = \frac{(O_i - E_i)^2}{E_i} + \frac{(n - E_i - O_i)^2}{n - E_i}
\] (5.5)

\[
\chi^2 = \frac{(n - E_i)(O_i - E_i)^2 + (E_i)(E_i - O_i)^2}{(E_i)(n - E_i)}
\] (5.6)

\[
\chi^2 = \frac{(n)(O_i - E_i)^2}{(E_i)(n - E_i)}
\] (5.7)

\[
O_i - E_i = \sqrt{\frac{\chi^2 \ast (E_i)(n - E_i)}{n}}
\] (5.8)

Moving the \(O_i\) to the right of the equation 5.8 shows that the value depends on the square root of the critical value, the number of samples, the expected number of the passing and failing units. Eight sites will be used to demonstrate the relationship in the equation 5.8. Therefore, the degree of freedom will be 7 and the critical value is 14.067 when \(\alpha = 0.05\) and 18.475 when \(\alpha = 0.01\).

Figure 5.23 shows the minimum change in the observed yield from the expected yield needed to reject the null hypothesis using equation 5.8. The Figure shows the data when the expected yield changes from 1% to 100% when the significance values are 0.05 and 0.01 and samples of 30 and 100. From the Figure, a minimum change of 21.7% in the yield will cause the \(\chi^2\) test to alert the operators when the expected yield of 90%, a significance level of 0.05, and 30 samples.

On the other hand, if the number of samples is 100 samples then a change of 11.81% in the yield will be sufficient to alert the operators. Statistically, the results from 100 samples are
more accurate than the 30 samples if the data is taken under the same conditions. Moreover, if the significance level changed to 0.01 then a minimum change in the yield is 24.55% for the 30 samples and 13.45% for 100 samples. It clear that when the significance level is 0.01 the change between the expected yield and real yield needed to reach the $\chi^2$ critical value is higher than $\chi^2$ critical value needed for significance level the 0.05.

Therefore, higher confidence can be reached using a larger number of samples or by increasing the confidence level. However, a larger number of samples will cause more time testing the units and it is important to flag any possible issue in the site as soon as possible to avoid the extra cost in retesting, scarping of good units as bad units or and damage to the hardware.

![Sensitivity with significance level = 0.05 and 0.01 and n = 30 and 100](image)

Figure 5.23. Sensitivity of 8 sites solution for different expected values with a significance value of 0.05 and 0.01 using 100 and 30 samples.
Figures 5.24 and 5.25 show the minimum yield change in the yield value to reject the null hypothesis and alert the operators as the number sites increase from 2 to 16 sites for a significance level of 0.05 and 0.01. The data show that as the number of sites increases the degree of freedom and minimum change in the yield increase too.

Figure 5.24. Sensitivity results of 2-16 sites solution for different expected values with a significance value of 0.01.

Figure 5.26 shows the real yield that will cause rejection of the null hypothesis if the expected yield 97% and a significance level of 0.05 when samples change from 30 to 100 samples. The Figure shows that as the number of samples increases the change in the real yield to reject the null hypothesis is less. Therefore, the real yield will be higher in the larger number of samples than the lower number of samples. However, that will take more units
Figure 5.25. Sensitivity results of 2-16 sites solution for different expected values with a significance value of 0.05.
Figure 5.26. Sensitivity of 8 sites solution for different number of samples and a significance value of 0.01.
Figure 5.27. Sensitivity of 8 sites solution for different number of samples and different expected yield with a significance value of 0.05.
to test where it will take 800 units in case of the 100 samples/site to run the $\chi^2$ test rather than the 240 units in the case of the 30 samples/site.

Finally, Figure 5.27 shows how the minimum change in the yield result to reject the null hypothesis will change due to the change in the expected yield and the number of samples/sites used. The higher the number of samples the lower the minimum change in the yield results. However, the minimum change in the yield is the highest when the expected yield is 50% and it best when it is close to 100% and 0%.
Chapter 6

Conclusion and Future Work

With the progress in the fabrication process, design tools, and the shrinking size of the transistors, there are hundreds of millions of transistors on some IC chips. Moreover, it is very common to have a system on chip (SOC). SOCs chips possibly could have radio-frequency circuitry, mixed-signal components, power management, digital signal processing, memories, and more functionality on the same chip. The cost per transistor of the chips has gone down due to the increase of the density when the transistor size shrinks. However, the reduced size has increased the number of circuits in the same area. Also, there is more integration and interconnection of the different circuits inside the chip with less observability of the signals within the chip.

All this means more blocks and signals to test, which increases the cost of testing. New methods are needed to reduce the cost to keep up with the design and process development. Those methods are fixed costs that are added to the chip. The extra cost depends on the total time of testing and the configuration of the tester used.

Going to high volume production increases the challenges regarding the test cost due to the variation in the load boards, the tester’s hardware configuration, the tester’s software revision, the prober, the handler, the materials, and the fabrications facility. Moreover, the continuous usage of test equipment stresses it and causes defects in the testers and the associated hardware. Those defects could cause the yield to diverge from the expected yield even when the actual yield of the parts themselves remains unchanged.

The real-time monitoring of the yield can save the chip manufacturer time and cost by finding issues with the testing system in the early stages of production (or as soon as those issues arise), avoiding the scrapping of good devices and preventing the penalty of sending bad units to the customer. A low-cost algorithm to monitor the yield of multiple test sites
is developed and debugged using data from production [74].

6.1. PaRent Testing Conclusion

The proposed PaRent testing reduces the test time by testing many units concurrently and running the different blocks simultaneously. The PaRent method is a very promising solution to reduce test cost, especially for high-volume, complex, mixed-signal devices. Although the PaRent method was not originally in the plan for the test solution for the device used in this case study, the throughput was boosted by more than 10% when various tests for other blocks were executed concurrently with the CC’s wait time.

Working on the PMIC chip to get the test time to the targeted value using PaRent allowed many issues to be defined and solved. The next section will discuss different tools that could make the process of conversion to PaRent testing easier.

6.1.1. Solution and Proposal for PaRent Testing for Mixed-Signal Devices

6.1.1.1. DFT Solutions for PaRent Testing

A well thought out DFT is a very powerful tool to speed up testing and reduce the complexity of load boards and ATE hardware [75]. DFT circuitry can be used for self-testing, especially in the case of memory, as in [76]. It can be used to aid in mixed-signal test, as in [77]. It is also used to test DACs and ADCs digitally, which helps save cost by testing the DAC and ADC together.

Testing the many blocks in the chips concurrently demands DFT that allows testing each block autonomously from the others. Usually, on SOC chips, there is a dependency between the blocks, where they share the same clock or the same reference voltage. So, the DFT should isolate the different blocks from the shared resources.

For the PaRent technique, the DFT needs to have the ability to turn on or off more than one block with a single write command or test mode setup. There is no need to keep track of the register values and test mode. However, this could lead to high inrush current
to fulfill the power demands of the different powered blocks if the blocks are not scheduled appropriately. If one is not careful, the total inrush currents of each site could exceed the current budget for the site. Moreover, the sum of those currents could exceed the maximum current of the tester’s limits.

Usually, many blocks share the same test net where the DFT is used to bring internal signals to a test pin to reduce the tester resources needed to test the chip to enable getting more sites tested in parallel. On the other hand, concurrent testing needs multiple test nets to shorten the test time by measuring the different outputs simultaneously. More test pins make the multiplexing less complex than connecting different channels to one test pin. Nevertheless, it will make it harder to find enough chip pins for each test pin.

One traditional solution to reduce pin count during the test is the use of scan chains, both internal scan chains and boundary-scan chains. Standardized solutions, such as IEEE 1149.1 JTAG [25] can enable the multiplexing of many internal scan chains, allowing digital results to be scanned and screened on the ATE. Furthermore, it provides access to a multitude of embedded instruments, including BIST engines [19], [20] and [21], sensors, hardware monitors, and configuration circuitry. Accessing many BIST engines and running BIST concurrently can significantly improve the concurrency of test (within the limits of the chip power, etc.), but today’s chips may include hundreds or thousands of embedded instruments, which would lead to very long scan chains and potentially complex instruction decoding of the JTAG instruction register (IR) contents in the Test Access Port (TAP) controller.

IEEE 1687 (IJTAG) gives better access to these embedded instruments [32]. IJTAG allows hierarchical grouping of instruments in dynamically reconfigurable scan chains and easy access. Unlike classic IEEE 1149.1, IEEE 1687 allows this reconfiguration to occur based entirely on data scanned through the test data register. Special scan cells, known as Segment Insertion Bits (SIBs), provide access to new scan chain segments when the correct value is clocked into the SIB’s update register on an UpdateDR (Update Data Register) cycle.
In brief, using IEEE 1687, one could theoretically group many memory BISTs (MBISTs) behind a SIB, open the SIB to place the MBISTs on the scan chain, shift data into the chain to put the MBISTs to start running, and then close the SIB to remove the interfaces from the chain. The MBISTs can then run concurrently, while the chain is shortened so that other instruments can be reached efficiently while the MBISTs are running. Thus, IEEE 1687 along with the embedded instruments is an important weapon in the arsenal of concurrent testing of internal blocks.

6.1.1.2. ATE software for Parent Testing

The software in the tester should have the capability to program various supplies to different levels and/or frequencies simultaneously. For example, different test pattern instruction encodings may be used to program the power supplies, analog supplies, RF supplies, and the digital pins on the ATE. The pattern format is vital. It can be used to set an instrument in the ATE to provide the right stimulus and capture the output at a precise time. For example, proper patterns can be implemented so that the DFT will reduce the inrush current to a suitable level at the right time. The test pattern should also allow multiple measurement instruments to be programmed concurrently. This will allow the instruments for all blocks in each site to be controlled by a single instruction and save test time. Some ATE companies already provide this capability to allow test patterns to control some of the analog instruments. However, it is not a common tool with all testers. It supports a small subset of the instruments in the tester. So, it is an important option to have when high volume mixed-signal devices are being tested.

Appropriate software is also needed to efficiently create, organize, and apply the patterns to be shifted into the chip to access embedded instruments in an IEEE 1687 [32] network when such a network is present in the die under test. Because the scan chain can be dynamically reconfigured in 1687, such software will need to keep track of which SIBs are currently open, which are closed, and which must be opened and/or closed or which ScanMux branches must be selected to access a particular set of embedded instruments. In general, concurrent access
to a set of test instruments in a well-designed IEEE 1687 network can be enabled simply by opening/operating the appropriate set of SIBs/ScanMuxes with a series of Shift Update operations. A sequence of iWrite and iRead commands, as described in the instruments’ PDL (Procedure Description Language) documentation can then be used to operate the instruments either individually or concurrently, depending on the needs of the test program. This test software should also be able to handle other test standards, such as IEEE 1149.1 and IEEE 1500 [78]. Such software is currently available for several Electronic Design Automation (EDA) and board test software companies.

Ideally, the ATE software should have the ability to easily enable/disable concurrent testing with some high-level software switches to ease the debugging process in cases of customer returns and initial development. For example, test software could be used to switch the test flow from concurrent to serial, where the test order of serial testing is specified in an excel sheet format or in a test variable read by the software.

Test software resources are also needed to help design the concurrent test flow itself. It is common to start with a serial flow program since it is easier to write and debug. However, going to concurrent flow from a serial flow is a challenging time-consuming task. The test time, the ATE resources used, and the device setup all need to be found for each block and test in the flow. The testing of each block should be scheduled in relation to other blocks without any conflict of resources or test set up. The sorting of the test time for each block to find the optimal order for running the test concurrently is also needed. Many researchers have investigated the test scheduling problem with respect to test time and test power consumption, and such algorithms may be useful here. Such a scheduling tool can be developed by the ATE companies, a third party, or in-house, but in any case, software aid is highly preferable to performing all scheduling by hand.

6.1.1.3. ATE software for PaRent Testing Future Work

As for parallel testing in a multi-site configuration, it is necessary to optimize the wait time in each block. Usually, each block has a wait time to allow settling of the tester’s
supplies and the device outputs. This wait time mainly depends on the number of sites running in parallel and the status of the test board. It is important to optimize the test wait time before going to concurrent mode.

Concerning wait time, the software tool’s main function is to find the minimum wait time without diminishing the yield for single-site and multi-site configurations. In the worst case, the test engineer can search for the minimum wait time by writing software that will iterate through multiple test applications with successively decreasing wait times. He then applies this test program to the device(s) on the tester. By observing the results of the test on each iteration of the for loop, the engineer can find the smallest wait time that yielded accurate test results. For complex chips, this ad hoc solution can be very time consuming because it requires new code to be written to find wait times for likely hundreds of different test measurements. The ATE company can help in cutting development time by providing this wait optimization coding capability in the software drivers for the different test instruments.

6.1.1.4. ATE and Load Board Hardware Requirements

In addition to the appropriate software, various hardware capabilities must also be present in the ATE to perform an efficient multi-site and concurrent test. In some cases, especially in low-cost testers, some of this functionality may be provided by an FPGA [79] on the ATE or on the load board that can be programmed to provide the missing functionality. For example, an FPGA could be used to enable the protocol-aware testing of I/Os.

The tester’s resources must also be spread evenly around the tester head to reduce the layout complexity when designing the load board. A clean layout and good working hardware are both essential for a high-volume production solution.

To test PMICs, such as the one used in this case study, it is important to have as many supplies as possible with at least 250mA load capability and up to 1A on the ATE. Many of the power management chips have multiple blocks with loads ranging from 150mA to 450mA. Furthermore, some supplies have a higher current capability because many of the PMIC’s SMPSs can go up to a 3A load. On the other hand, for the boost converter, which is
a DC-DC converter with an output voltage greater than the input voltage, the supply must be capable of sourcing a higher current than the load current.

Note that not all semiconductors will need all of these functionalities in ATE. Ideally, the function described here could be supplied as an option. The cost of the final product with and without those options is the main factor in determining whether to buy those options.

6.2. Real-Time Yield Monitor Conclusion

A real-time monitor of the yield and the quality of the test results in the production test can significantly decrease the time required to discover problematic sites, handlers, probers or wafers in a multi-site environment. The faster a problem is detected, the faster it can be fixed, and the less money is lost in retesting, recalling bad units, and scrapping good units. Thus, it is desired to develop a process of quickly recognizing a problem that can affect the yield at any site using easy, inexpensive adjustments to the test setup. The yield monitoring algorithm presented here examines the yields of each site and compares them to the expected yield using the $\chi^2$ test and produces an alarm if the yield is statistically dependent on the site or the tester used to test the parts.

The proposed algorithm can catch an issue and indicate whether it affects all the sites equally (e.g. due to aging) or if it affects only one site at a time. In the algorithms described here, an alarm is sent when a potential issue is detected, and a threshold of two failing windows is used to drive action, such as pausing the testing to circumvent any possible damaging consequences on the hardware or the units under testing. The threshold and the action taken could be changed depending on the type of the device, customer commitment, the production engineer’s experience, and the history of the device. For some parts and production environments, a pause of the production is favored over the common response of immediately scrapping the yield loss. For example, in the case of high current devices, it is a better solution to pause the production instead of possibly damaging the devices, test board, or tester.
The proposed algorithm [74] can be used for both probe testing of the wafer level units and for the final test that tests the packaged units. Thus, it has advantages over procedures that are only used for the wafer-level test. Also, the number of failing units in the algorithm is smaller than the number of failing units in the post-processing method. This will reduce the time of retesting the questionable units.

The suggested algorithm is also easy to implement as a real-time monitor in the test program of standard testers used in the semiconductor industry without the need for new infrastructure or significant expenditures. There are companies in the semiconductor testing industry that sell services to monitor the yield and send reports about the yield almost in real-time. It is almost in real-time since the data has to be sent over the network in the Standard Test Data Format (STDF) [39]. Another issue is that the STDF files are available only after stopping the test when the lot is done. While such services can provide detailed and valuable information for yield learning and addressing other processes and test problems, the proposed approach can provide a simple solution that can be employed to identify problems with the tester at a very low cost if implemented early in production program if outside services are not contracted.

The time and training required to set up the code are likely to be minimal, and a library could be prepared by the test engineer and reused with every new program. The \( \chi^2 \) test could also be appended to the operating system of the tester. This would reduce the complexity of the code and make it faster to implement for new devices.

The \( \chi^2 \) test is standard in many statistical software packages. Fast algorithms for its computation exist in several computer languages.

6.3. Real-Time Yield Monitor Future Work

For future work, this technique can be implemented in a setting where the program will run the load board diagnostic program and tester diagnostic program, if possible, for the board before sending the email and attaching the diagnostic reports with the email to isolate the problem to a load board issue, tester issue or device issue.
Finally, keeping track of the different issues with their symptoms can be logged and tracked to address any future similar issues. The process of addressing future issues can be manual. However, automating the process and using Artificial Intelligence (AI) to address the issues for that specific chip or similar types of chips could potentially make the process more accurate and efficient.
Appendix A

Statistical Background

Statisticians follow a formal process, called hypothesis testing, to determine whether to reject a null hypothesis based on sample data.

A.1. Decision Errors

Two kinds of errors happen in hypothesis testing. A Type-I error is a false positive error where the true null hypothesis is rejected incorrectly. On the other hand, a Type-II error is a false negative where a false null hypothesis is retained incorrectly. The probability of a Type-I error occurring is denoted by $\alpha$ and is the probability of rejecting the null hypothesis when it is true. The probability of a Type-II error occurring is denoted by $\beta$ and it is the probability of rejecting the alternative hypothesis when it is true. Table A.1 shows $1 - \alpha$ and $1 - \beta$ and they represent making the correct decision. $1 - \beta$ is called the power of the test.

Table A.1. Table of error conditions.

<table>
<thead>
<tr>
<th></th>
<th>Null hypothesis ($H_0$) is TRUE</th>
<th>Alternative hypothesis ($H_1$) is TRUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rejected</td>
<td>$\alpha$</td>
<td>$1-\beta$</td>
</tr>
<tr>
<td>Not Rejected</td>
<td>$1-\alpha$</td>
<td>$\beta$</td>
</tr>
</tbody>
</table>

A.2. Pearson’s Chi-squared test

Pearson’s Chi-squared test is used to evaluate three types of tests: homogeneity, goodness of fit, and independence. The three types are explained as next:
1. A test of goodness of fit confirms if an observed frequency distribution deviates from a theoretical distribution or not.

2. A test of homogeneity examines the distribution of counts for two or more groups utilizing the same variable.

3. A test of independence evaluates if two categorical variables are significantly associated.

The Pearson’s Chi-square test \( \chi^2 \) calculates the deviation of the data from the expected value as using Equation

\[
\chi^2 = \sum_{i=1}^{i=n} \frac{(O_i - E_i)^2}{E_i}
\] (A.1)

where \( O_i \) is the observed frequency in the contingency table for cell \( i \), \( E_i \) is the expected frequency of cell \( i \) assuming that the null hypothesis is true, and \( n \) is the number cells in the contingency. If there are at least five observations within each cell of a contingency table of the data, then equation A.1 follows a \( \chi^2 \) distribution with \( (r-1)(c-1) \) degrees of freedom, where \( r \) is the number of rows in the table and \( c \) is the number of columns. The \( \chi^2 \) distribution is shown in A.1.

A contingency table or association table is used when the samples are classified in two different ways. It is built by tabulating the actual frequency distribution of the variables in the observed samples. For the \( \chi^2 \) test, the measurements should not represent continuous measurements since \( \chi^2 \) is applied to discrete data only. Thus, a data set of five categories, where each category has two levels, can be represented in a contingency table of five columns by two rows.

Table A.2 represents the observed value indicated as L1 (Observed) and L2 (Observed) in the first line of each cell. The second line in the cell shows the calculation of the expected value as L1 (Expected) and L2 (Expected). Similarly, R1 represents the sum of the observed values in row one and R2 represents the sum of the observed values in row two. C1 and C2 represent the sum of the observed values for their respective columns. \( A_{11} \) is the observed data for level 1 and category 1, \( A_{12} \) is the observed data for level 1 and category 2, \( A_{21} \) is
Figure A.1. Chart of $\chi^2$ distributions.

Table A.2. Contingency table of two categories and each category has 2 levels with the observed value and the expected value for each level and category.

<table>
<thead>
<tr>
<th>Category 1</th>
<th>Category 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1(Observed)</td>
<td>$A_{11}$</td>
<td>$A_{12}$</td>
</tr>
<tr>
<td>L1(Expected)</td>
<td>$R_1 \times C_1 / M$</td>
<td>$R_1 \times C_2 / M$</td>
</tr>
<tr>
<td>L2(Observed)</td>
<td>$A_{21}$</td>
<td>$A_{22}$</td>
</tr>
<tr>
<td>L2(Expected)</td>
<td>$R_2 \times C_1 / M$</td>
<td>$R_2 \times C_2 / M$</td>
</tr>
<tr>
<td>Total</td>
<td>$C_1 = A_{11} + A_{21}$</td>
<td>$C_2 = A_{12} + A_{22}$</td>
</tr>
</tbody>
</table>
the observed data for level 2 and category 1, and $A_{22}$ is the observed data for level 2 and category 2.

Next is an example of tossing two coins twenty times to explain the $(\chi^2)$ test for independence test. Table A.3 describes the experiment of tossing two coins twenty times and keeping the tally of the number of heads and tails. The table represents that the first coin produced twelve heads and eight tails. On the other hand, the second coin produced fourteen heads and six tails.

Table A.3. Contingency Table of tossing two coins ten times showing the observed values and the expected values.

<table>
<thead>
<tr>
<th></th>
<th>Head</th>
<th>TAIL</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1(Observed)</td>
<td>12</td>
<td>8</td>
<td>Row 1 =20</td>
</tr>
<tr>
<td>L1(Expected)</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>L2(Observed)</td>
<td>9</td>
<td>11</td>
<td>Row 2 =20</td>
</tr>
<tr>
<td>L2(Expected)</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>Column 1 = 21</td>
<td>Column 2 = 19</td>
<td>M = 40</td>
</tr>
</tbody>
</table>

The expected value in $\chi^2$ is calculated using the independence of the data under the null hypothesis. Under the independence theory, if $A$ and $B$ are independent then

$$P(A \cap B) = P(A) \times P(B)$$

(A.2)

where $P(A)$ is the probability of event $A$ occurring and $P(B)$ is the probability of event $B$ occurring. $P(A \cap B)$ is the probability of the intersection of events $A$ and $B$ which means $A$ and $B$ will happen. This is used to calculate the expected values in the contingency table as shown in table A.2.

After calculating the $(\chi^2)$ statistic using Equation A.1, p-value for the test is found using the $\chi^2$ tables using the degrees of freedom of the data.

Table A.5 shows the critical $\chi^2$ value for different degrees of freedom for significance
levels of 0.01 and 0.05. The Excel function CHIINV was used to create Table A.5. As discussed earlier regarding the $\chi^2$ test, the critical value will be used to reject or accept the null hypothesis, where the null hypothesis tests results which are independent of the site used to test the part.

The p-value is the probability of obtaining a test statistic as extreme or more extreme than the one obtained from the data if the null hypothesis is true. Therefore, the smaller the p-value, the stronger the evidence is against the null hypothesis of independence. It used to be that the p-value is declared as small if it is smaller than the stated significance level of the test. The significance level is given before the test is conducted and it cannot be changed once it is set. Rejecting the null hypothesis indicates that the data do not support the hypothesis of independence. It is important to know that the American Statistical Association recommended to stop using the term “statistical significance” and to report the p-value and the confidence limits with their respective definitions as evidence for or against the null hypothesis.

Table A.4 shows the $\chi^2$ calculation for the data of the coin tossing experiment. It also has the probability found in the $\chi^2$ table for $\chi^2$ and d.f. = 1. The table shows the probability is 0.045. In other words, there is a 1 in 22 chance of obtaining a chi-square test statistic value
Table A.5. Critical $\chi^2$ values for significance level of 0.05 and 0.01 for different degree of freedom calculated using Excel.

<table>
<thead>
<tr>
<th>Degree of Freedom</th>
<th>$X^2$ for significance level = 0.05</th>
<th>$X^2$ for significance level = 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5.99</td>
<td>9.21</td>
</tr>
<tr>
<td>3</td>
<td>7.81</td>
<td>11.34</td>
</tr>
<tr>
<td>4</td>
<td>9.49</td>
<td>13.28</td>
</tr>
<tr>
<td>5</td>
<td>11.07</td>
<td>15.09</td>
</tr>
<tr>
<td>6</td>
<td>12.59</td>
<td>16.81</td>
</tr>
<tr>
<td>7</td>
<td>14.07</td>
<td>18.48</td>
</tr>
<tr>
<td>8</td>
<td>15.51</td>
<td>20.09</td>
</tr>
<tr>
<td>9</td>
<td>16.92</td>
<td>21.67</td>
</tr>
<tr>
<td>10</td>
<td>18.31</td>
<td>23.21</td>
</tr>
<tr>
<td>11</td>
<td>19.68</td>
<td>24.72</td>
</tr>
<tr>
<td>12</td>
<td>21.03</td>
<td>26.22</td>
</tr>
<tr>
<td>13</td>
<td>22.36</td>
<td>27.69</td>
</tr>
<tr>
<td>14</td>
<td>23.68</td>
<td>29.14</td>
</tr>
<tr>
<td>15</td>
<td>25.00</td>
<td>30.58</td>
</tr>
<tr>
<td>16</td>
<td>26.30</td>
<td>32.00</td>
</tr>
<tr>
<td>17</td>
<td>27.59</td>
<td>33.41</td>
</tr>
<tr>
<td>18</td>
<td>28.87</td>
<td>34.81</td>
</tr>
<tr>
<td>19</td>
<td>30.14</td>
<td>36.19</td>
</tr>
<tr>
<td>20</td>
<td>31.41</td>
<td>37.57</td>
</tr>
</tbody>
</table>
of 4 if the null hypothesis is true. However, the example did not specify the significance level as it should be. In most of the data analysis, 5% and 1% are the significance levels used in many of the statistical data analysis.

For the above example, the p-value is 0.045 which is less than the prespecified level of $\alpha = 0.05$. Therefore, the hypothesis that our coin is fair is rejected using the $\alpha = 0.05$ as a significance level.
BIBLIOGRAPHY


