

國立交通大學

電子工程學系 電子研究所

博士論文

EDA 和測試方法之於先進 CMOS 製程技術變異的特性
分析與降低化



EDA and Testing Methodologies for Characterization and
Minimization of Process Variation on Advanced CMOS
Technology Nodes

研究生：羅增錦

指導教授：趙家佐 博士

中華民國一百零一年五月

EDA 和測試方法之於先進 CMOS 製程技術變異的特性
分析與降低化

EDA and Testing Methodologies for Characterization and
Minimization of Process Variation on Advanced CMOS
Technology Nodes

研究生：羅增錦
指導教授：趙家佐

Student : Tseng-Chin Luo
Advisor : Mango C.-T. Chao



Submitted to Department of Electronics Engineering and
Institute of Electronics
College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Electronics Engineering

Apr 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年五月

EDA 和測試方法之於先進 CMOS 製程技術變異的特性 分析與降低化

學生：羅增錦

指導教授：趙家佐

國立交通大學

電機學院

電子工程學系

電子研究所

摘要

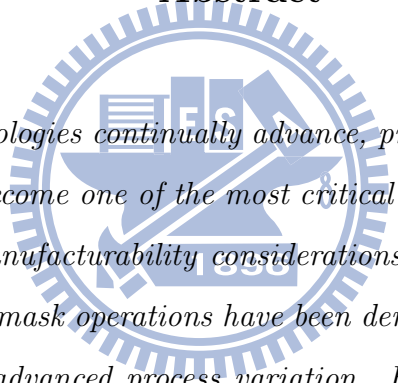
隨著製程技術持續推新進步，製程變異急遽增加並漸漸成為 IC 製造最關鍵的因素之一。基於 design-for-manufacturability 的考量，EDA 方法中像是 dummy fill 和 boolean mask operations 等在縮小先進製程變異上都已被驗證是一種很有效率的技術。然而，在新的先進技術設計上想在首次 tape-out 便能成功，必須避免過長的處理時間和不夠縝密的驗證流程。另外從製程開發的角度而言，在先進製程開發階段，常會利用 array-based test structure design 來量測大量電晶體 DUT 以監控制程的變異，尋求具統計性、關鍵性的結論。但起因於控制電路影響而造成的測試時間過長及測試結果不準確都會阻礙先進製程開發的挑戰。本論文從 EDA, array-based test structure design 到參數測試領域提出數種技術來縮小製程變異和改善晶圓設計驗證流程。除此之外，本論文也針對相應的特性測試分析，快速和高準確度的測試方法進行了多方驗證。

EDA AND TESTING METHODOLOGIES FOR CHARACTERIZATION AND MINIMIZATION OF PROCESS VARIATION ON ADVANCED CMOS TECHNOLOGY NODES

Student: Tseng-Chin Luo Advisor: Dr. Mango C.-T. Chao

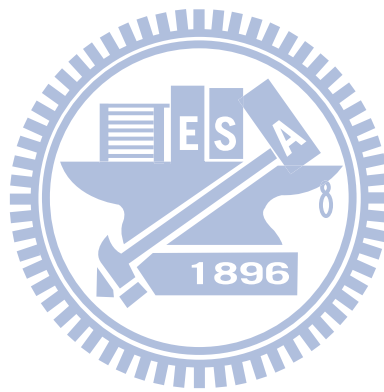
Department of Electronics Engineering & Institute of Electronics
National Chiao Tung University

Abstract



As process technologies continually advance, process variation has greatly increased and gradually become one of the most critical factors for IC manufacturing. Based on design-for-manufacturability considerations, EDA methodologies such as dummy fill and boolean mask operations have been demonstrated to be effective techniques for minimizing advanced process variation. However, long processing time and insufficiently robust verification flows are significant obstacles for delivering functional first silicon on new advanced technology designs. From the process development point of view, to obtain statistically significant and conclusive results, a large number of DUTs using an array-based test structure design is a commonly used technique for variation monitoring at the advanced process development stage. However, long testing time and inaccurate test results due to the influence of the control circuitry pose further challenges for advanced process technology development. This thesis presents several techniques from EDA to parametric testing to minimize

process variation, and enhance the verification flow. In addition, the corresponding characterization methodology using an array-based test structure design and a fast and highly accurate testing methodology have been also demonstrated in this work.



Acknowledgements

There are many people to thank, but it is not difficult to know where to begin this list. I thank my parents, who both gave me life and taught me how to live it. My wife, Elain Lai, has been a wonderful companion over the past 10 years, sharing many happiness and hardships, and providing a constant source of warmth and support.

At NCTU, my advisor, Dr. Mango C.-T. Chao, provided both academic and personal guidance from the day I walked into his laboratory until the day I stood before my committee and defended this work. I credit him with showing me how to attack problems which seemed larger than I am, and to keep the big picture in mind when the details seemed overwhelming. Prof. Chao was a great source of both testing insight and perspective on life. His challenge to attempt an ab-initio understanding of my experiment provided much of the motivation for the writing of this thesis, for which I am grateful.

To all of you, I thank you from the bottom of my heart. I could never have done it without you, and I hope one day I can at least give back a fraction of what you have given me.

Tseng-Chin Luo

National Chiao Tung University

June 2011

Table of Contents

Abstract (Chinese)	i
Abstract	ii
Acknowledgements	iv
List of Tables	ix
List of Figures	x
Chapter 1. Introduction	1
Chapter 2. A Novel Design Flow for Dummy Fill Using Boolean Mask Operations	4
2.1 Background	10
2.1.1 Long-range vs. Short-range Dummy Patterns	10
2.1.2 Dummy-Pattern Style	11
2.1.3 Computation of Mask Layers Using Boolean Operations	12
2.2 Boolean-Operation Dummy Fill	12
2.2.1 Overall Flow	12
2.2.2 Detailed Steps of Boolean-operation Dummy Fill	14
2.2.3 Dummy Fill with Long- & Short-range Dummy Patterns	20
2.2.4 Dummy Cell Initial Definition For Optimizing Layout Density Uniformity	24
2.3 Experimental Results	26
2.3.1 Conventional Dummy Fill & Test Cases	26
2.3.2 Comparison of Runtime and GDS File Size	26

2.3.3	Chip Layout after Dummy Fill	31
2.3.4	Pattern Density after Dummy Fill	33
Chapter 3. Mask versus Schematic – An Enhanced Design-Verification Flow for First Silicon Success		35
3.1	Background	38
3.1.1	Layout versus Schematic	38
3.1.2	Mask Boolean Operation	39
3.1.3	Customized device using mask boolean operation	40
3.2	Mask Versus Schematic Methodology	42
3.2.1	Overall Flow	42
3.2.2	Virtual Mask Sets Generation	45
3.2.3	MVS Runset	47
3.3	Experimental Results	50
3.3.1	Boolean Operation Error	51
3.3.2	Braces Placement Error	52
3.3.3	Sizing Error	53
3.3.4	Qualification of MVS	56
3.3.5	IP design experiment	56
Chapter 4. A novel array-based test methodology for local process variation monitoring		57
4.1	Background	60
4.1.1	Traditional Testline(PCM)	60
4.1.2	Transistor Array Test Structures and Adaptive Voltage Compensation	62
4.1.3	ROM-like transistor array	65
4.2	Design Methodology	66
4.2.1	Design Architecture	66
4.2.2	Hardware IR Compensation	68

4.2.3	Voltage Bias Elevation for Measuring Ion	71
4.2.4	Leakage-Current Cancellation for Measuring I_{off}	73
4.3	Experimental Results	78
4.3.1	Proposed Test Structure vs. Traditional PCM Testline	78
4.3.2	Effectiveness of Hardware IR Compensation	80
4.3.3	Effectiveness of Leakage-Current Cancellation	81
4.3.4	Proposed Array vs. ROM-like Array	81
4.3.5	Hardware IR compensation v.s Adaptive Voltage compensation . .	82
4.3.6	Local Mismatch Measured by Proposed Test Structure	83
 Chapter 5. Fast transistor threshold voltage measurement method for high-speed, high-accuracy advanced process characterization		93
5.1	Background	96
5.1.1	Measuring constant current V_t using a binary search algorithm . .	96
5.1.2	Improving constant current V_t testing time using an interpolation methodology	98
5.1.3	Fast V_t measurement using an on-chip operational amplifier based test structure	100
5.2	Design Methodology	104
5.2.1	OP-based SMU for Fast V_t Measurement	104
5.2.2	Implementation for Stand-alone DUT	106
5.2.3	Implementation for Array Test Structure V_t measurement	108
5.3	Experimental Results	108
5.3.1	Binary search V_t testing time	108
5.3.2	V_t testing time improvement using the interpolation method . . .	111
5.3.3	Simulation of V_t measurement using OP-based SMU	112
5.3.4	Stand-alone DUT test result	114
5.3.5	Array-based test structure result	115

Chapter 6. Conclusion	118
Bibliography	120
Vita	133
Publication List	134



List of Tables

2.1	Resulting GDS file size and runtime for conventional dummy fill.	27
2.2	Runtime of the proposed boolean-operation dummy fill.	30
2.3	Resulting GDS file size and runtime for the proposed boolean-operation dummy fill.	31
3.1	MVS statistics on a IP circuit with about 3.5 million transistors.	56
4.1	Comparison between hardware IR compensation and adaptive voltage compensation.	83
5.1	Comparison of device characteristics and testing time for Vt measurement using binary search for two different transistors. The different test times result mainly from the different magnitudes of the target current.	110
5.2	Test time comparison of OP-based Vt measurement between stand-alone DUT and array-based DUT.	117

List of Figures

2.1	Conventional dummy fill vs. proposed dummy fill. Units of t.u. are arbitrary units of time, on the order of 24 hours, though precise execution times of course vary for different facilities.	7
2.2	Example of different dummy-pattern styles.	11
2.3	Steps of the proposed boolean-operation dummy fill.	16
2.4	Dummy Tile Design.	16
2.5	An example of the jog removal.	19
2.6	Example of boolean-operation dummy fill using only short-range dummy patterns.	21
2.7	Example of boolean-operation dummy fill using both short-range and long-range dummy patterns.	24
2.8	Definition of dummy cells for different chip sub-regions based on initial layout density	25
2.9	Illustration of disruption of the GDS hierarchy of a dummy pattern array after applying one boolean operation	31
2.10	A portion of the layout of the test case Logic1 after performing the boolean-operation dummy fill.	32
2.11	Layout of the test case "Memory" after applying conventional dummy fill and boolean-operation dummy fill respectively.	32
2.12	Surface and contour plot of pattern density after performing the traditional and proposed boolean dummy fill.	34
3.1	Comparison of layout pattern change due to various operations commonly applied to completed design GDS.	37
3.2	Conventional Flow from Design to Mask Making.	39

3.3	Mask layer formation by boolean operation. In general, P Well and NLDD layers are generated by mask boolean operations. Other layers are directly copied with only sizing operations.	41
3.4	Comparison of LDD mask layers generated by correct and erroneous mask boolean operation equations.	41
3.5	Customized device formation by boolean operation. The ultra high Vt device is generated by introducing a new layer and mask boolean operation.	42
3.6	MVS Flow for Mask Generation Algorithm Verification.	44
3.7	Virtual mask set generation flow.	45
3.8	LVS TEXT Layers need to be copied to the virtual mask set GDS to facilitate net recognition and debugging.	46
3.9	MVS runset generation flow using a simplified transistor formation example. The only difference between the GDS used for MVS and LVS is the presence of the NLDD layer in the MVS database. Real cases are much more complicated, as they contain many more additional layers generated by the mask generation algorithm.	47
3.10	Resistor terminal contact placement in tapeout GDS.	49
3.11	Sizing operations performed on the Poly and AA layers cause the gate dummy ID layer to misalign with the biased gate.	50
3.12	Correct mask algorithm for implant layer and the corresponding virtual mask set GDS with the generated implant layer.	51
3.13	Incorrect boolean operation in mask algorithm and the resulting missing implant layer.	51
3.14	MVS error from incorrect Boolean operation. Column 1 lists devices correctly matched between layout and schematic. Column 2 lists schematic devices not matched to any layout device. Column 3 lists layout devices which were not matched to any schematic device, and column 4 lists the respective device names of all matched and unmatched devices. The last 2 rows show the total number of devices and the total number of matched and unmatched nets respectively.	52
3.15	Correct mask algorithm for implant layer and the corresponding virtual mask set GDS with no extra generated implant layer.	53
3.16	Incorrect braces placement in boolean operation and the resulting additional implant layer.	53

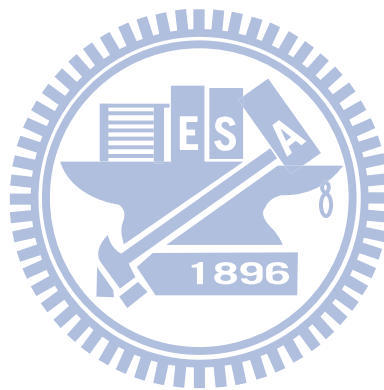
3.17	MVS error resulting from braces placement error.	54
3.18	Equation shows the correct mask algorithm for implant mask generation and the virtual mask set GDS with the correctly sized implant layer. . . .	54
3.19	Incorrect sizing value in Boolean operation creates an implant layer shape layer smaller than corresponding transistor AA.	55
3.20	MVS error from sizing value error.	55
4.1	The configuration of a conventional PCM testline.	62
4.2	Voltage compensation mechanism used in a conventional PCM testline. . .	62
4.3	The schematic of the adaptive voltage compensation for one DUT.	64
4.4	Pseudo code of adaptive voltage compensation algorithm for Ion measurement.	64
4.5	The architecture of a ROM-like transistor array using shared common gate and drain buses [80].	66
4.6	Layout of a ROM-like DUT array using straight poly lines for gate connections [80].	66
4.7	Architecture of the proposed transistor array with 4x64 DUTs for scribe-line-compatible footprints.	68
4.8	Layout of the proposed test structure using circuit-under-pad design (CUP) for scribe-line compatible footprints. (a) cross-section view of CUP design, (b) top view of the proposed test structure, and (c) schematic inside each test unit.	69
4.9	Schematic of the proposed hardware IR-compensation mechanism.	71
4.10	An example of the negative node created without applying the proposed voltage bias elevation technique.	72
4.11	No forward biased current on the transmission gate is generated after applying voltage bias elevation ($V_{elv}=0.5V$).	74
4.12	Possible leakage paths in the proposed array-based test structure. The thick line indicates the selected path. The background leakage current includes the leakage from both the selected and unselected paths.	75
4.13	Larger background leakage due to unbalanced leakage paths before applying voltage bias elevation.	76

4.14	Reducing the leakage current by balancing the leakage paths with optimized voltage bias elevation.	76
4.15	Background leakage reduction by offset voltage from half V_{DD} for the 64 DUTs in the array for 1 wafer (9 die/wafer with 1 array/die). . . .	78
4.16	I_d versus V_g measured by (A) a PCM testline and (B) a proposed array-based test structure with log scale (upper figure) and linear scale (lower figure).	86
4.17	Locations of measured test structures.	87
4.18	I_{on} and V_{th} of each DUT measured by 9 PCM testlines and 9 proposed test structures. The curve is an elliptical fit of the 95% confidence interval of the data set.	87
4.19	I_{on} and V_{th} of each DUT measured by only one PCM testline and one proposed test structure. The curve is an elliptical fit of the 95% confidence interval of the data set.	88
4.20	V_{th} vs DUT column (X axis) for all four rows of the structure, with each row shifted by a fixed offset along the Y axis.	88
4.21	I_d versus V_g measured by (A) a PCM testline, (B) a proposed array-based test structure and (C) a proposed array-based test structure without applying hardware IR compensation.	89
4.22	I_{on} and V_{th} of each DUT measured by proposed test structures with and without the hardware IR compensation. The curves are elliptical fits of the 95% confidence intervals of the two data sets.	89
4.23	I_{on} vs. I_{off} of each DUT measured by the proposed test structures with and without applying the current-cancellation technique.	90
4.24	I_d versus V_g measured by a ROM-like DUT array and a proposed DUT array.	90
4.25	I_{on} vs. V_{th} of each DUT measured by a ROM-like DUT array and a proposed DUT array.	90
4.26	Layout of a paired-transistor DUT.	91
4.27	V_{th} mismatch of paired MOSFETs measured by the proposed test structures for different W/L dimensions.	91
4.28	Poly-CD mismatch of paired MOSFETs measured by scanning electron microscope for different W/L dimensions.	92

4.29	V_{th} vs poly CD for all MOSFETs in the array of MOSFET pairs.	92
5.1	Voltage compensation mechanism used in a conventional PCM testline. . .	98
5.2	SMU connection and bias condition for V_t measurement using a binary search approach for n-FET.	99
5.3	Pseudo code of binary search V_t measurement.	100
5.4	Example of iterating V_g to obtain drain current matching target within specified criteria.	101
5.5	Setting the initial gate voltage values to interpolate to the gate voltage corresponding to the target drain current. Log scale is used only for ease of visualization. In practice, linear scale is used for V_t interpolation. . . .	102
5.6	Double hump transistor I-V curve resulting from use of STI.	102
5.7	Only a single force-measure iteration is required by the OP-based V_t measurement technique.	103
5.8	Circuit schematics for OP-based V_t measurement, where the DUT is (a) n-FET, and (b) p-FET.	104
5.9	The configuration of an OP-based SMU and a n-FET for V_t measurement with one force-measure iteration. The target current defined for V_t is forced as a negative current by an additional SMU.	106
5.10	SMU connections for fast V_t measurement by using the proposed OP-based SMU	107
5.11	Pseudo code of V_t measurement by OP-based SMU.	107
5.12	V_t measurement by OP-based SMU in an Array Test Structure.	109
5.13	Time trace of successive iterations for V_t measurement by binary search, showing gate voltage (left axis) and measured current matching percentage (right axis). I_d is the measured current at the drain node and I_t is the target current for V_t definition by the constant current criteria.	111
5.14	Scatter-plot of V_t s measurements by the binary search vs the interpolation method.	112
5.15	Simulation of V_t measurement by an OP-based test structure. Plotted data are (a) OP output voltage, i.e. V_t , and (b) DUT source voltage, which is clamped at 0V due to virtual short to V_{set}	113

5.16 Transient simulation of OP-based V_t measurement in an array-based test structure 114

5.17 V_t distribution obtained by 1000 repeated measurements on the same DUT (shorter channel transistor) by OP-based measurement of the array test structure. 115



Chapter 1

Introduction

As process technologies continually advance, process variation has greatly increased and gradually become one of the most critical factors for IC manufacturing. Several techniques such as dummy fill, pattern correction by boolean operation have been demonstrated to be an effective technique to reduce process variation and improve manufacturability for advanced IC designs. However, the computation load, often several days for a realistic IC design, is a significant portion of the cycle time for delivering first silicon on new or modified designs.

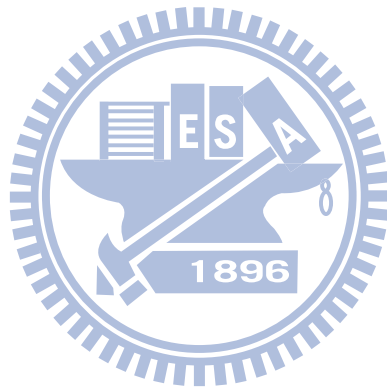
In addition, Layout versus Schematic (LVS) is a commonly used technique employed at the design stage to insure the correctness of physical layout. However, as process technologies continually advance, design layout which has been verified by LVS may undergo substantial layout changes when subjected to the mask generation booleans, with potential implications for performance and margin estimation, particularly given the aggressive use of stressors in modern CMOS technologies. Errors in mask generation booleans, which are very difficult to detect by present primitive inspection methods, can easily result in functional failure although the initial LVS predicted success. Therefore, LVS performed at the design stage is no longer an iron-clad guarantee of chip functionality in advanced process technologies.

Moreover, to monitor local process variation, a large number of DUTs (devices-

under-test) in close proximity must be measured. Array-based test structures design also has been widely employed in advanced process development for process variation monitoring. However, the measurement accuracy and testing time have always been a big concern for massive measurement.

In this paper, first of all, we propose a novel design flow and dummy-fill algorithm based on Boolean operations, which greatly improves computational efficiency and pattern density uniformity, and enables dummy generation to be combined with the mask-preparation Boolean operations performed by the mask fabrication facility. Mask data preparation can be performed in parallel with dummy generation and post-dummy simulation checks at the design house, resulting in improved first-silicon cycle time. Experimental results demonstrate these benefits in the context of an advanced foundry process technology. Second, we introduce Mask-versus-Schematic (MVS) verification, a novel design verification flow which directly compares the schematic netlist with a netlist extracted after application of all mask generation booleans, in order to insure the correctness of the final mask data just before tape-out. Furthermore, the introduced methodology can be performed using currently available physical verification EDA tools. The experimental results presented here, using examples from some of the industry's most advanced process technology nodes, demonstrate the effectiveness and efficiency of this methodology in detecting errors resulting from mask generation boolean operations. Third, several techniques can guarantee high measurement accuracy in array-based test structures by application of the test techniques proposed in this paper: hardware IR compensation, voltage bias elevation, and leakage-current cancellation. Finally, a very fast threshold voltage measurement methodology by utilizing Operational Amplifier-Based(op-amp) SMU test configuration has been demonstrated. The test methodology not only greatly improves test speed but also testing efficiency. All of proposed methodolo-

gies in this paper resulting from the few most advanced process-technology nodes demonstrate the effectiveness and efficiency on testing and minimization for process variation.



Chapter 2

A Novel Design Flow for Dummy Fill Using Boolean Mask Operations

As process technologies continually scale, the process variation due to each process step has greatly increased and gradually become one of the most critical factors for IC manufacturing. In order to reduce the level of process variability, several physical-design techniques have been proposed to constrain or modify the original IC layout for improved manufacturability. Such design-for-manufacturability (DFM) techniques include: (1) symmetry or matching constraints [1] [2], which can reduce the potential voltage offset and power-supply rejection ratio for analog circuits, (2) via doubling [36] [37] and wire spreading [38] [39], which can lower the impact of random defects by adding redundancy, (3) micro-regularity constraints, such as single orientation of critical dimension (CD) lines and constant poly pitch, for RET (resolution enhancement technique) compatibility [7], and (4) dummy fill, which inserts dummy layout patterns (which are not electrically active) into the design layout to improve pattern uniformity and thereby improve the uniformity of planarization, stress balance, and thermal anneal. In this paper, we will focus on improvements to the dummy fill algorithm and its use in the design flow.

Conventional dummy fill requires two steps. The first step is to define the layout pattern of *dummy cells* [8] [9] [10] [11], which usually follows the guidance of IC foundries based on their process characteristics. The second step is the *dummy-*

cell placement, which places the defined dummy cells into the empty space of a design layout to balance its feature density. Several algorithms for dummy-cell placement have been proposed according to various density models, such as ILD (inter-layer dielectric) CMP models [12] [13] [14], Cu CMP models [15] [16], and a hybrid model combining both CMP and electroplating (ECP) topography models [17].

Figure 2.1(a) shows the conventional design and tape-out flow, in which the dummy fill is applied by the design house after the design layout has passed DRC and LVS, resulting in the final GDS file to be delivered to a IC foundry for tape-out (hereafter, the "tape-out GDS file"). After the dummy cells are placed into the layout, a final simulation verification (the post-dummy simulation step) is performed, and the tape-out GDS file is sent to an IC foundry, which then performs the following steps: (1) computation of the corresponding mask layers, (2) visual inspection of the mask layer data, commonly referred to as "Jobview", and optical proximity correction (OPC), and (3) generation of the reticle set for wafer processing. Traditionally, particularly for technologies up to 65nm, the post-dummy simulation is generally used only as a safety verification check to assess small changes in the design performance due to parasitics and layout-dependent effects introduced by the dummy patterns. Unless serious timing impact is observed, the results of this simulation very rarely drive modifications to the original design before tape-out because repeated iterations of layout-change, dummy-fill, and simulation are costly in terms of both design cycle time and computational overhead. In this design flow, although the design is known to be functional after DRC and LVS, before wafer processing can begin, several days must be spent on dummy generation and post-dummy simulation. Furthermore, the GDS file size increases dramatically with the addition of dummy patterns, often by a factor of 10 or more, particularly for advanced process technologies. This large GDS file is often difficult to handle, and

dummy generation by typical algorithms and post-dummy simulation is often a large computational load for the workstations used by a typical design house for DRC and LVS.

It should be noted that although generally all design modifications are based on performance simulations before dummy-fill, for high performance designs in advanced technologies below 65nm, the parasitic impact of dummy patterns on product performance may potentially be large enough to require at least minor layout modification, and possibly one or more iteration of layout change, dummy-fill, and simulation. Although such cases are relatively rare in the authors' experience, if multiple iterations of layout change and post-dummy-simulation were required, the benefits of our proposed algorithm would in fact be multiplied because each iteration would benefit from the improved computational efficiency and database hierarchy. However, in order to make our discussion applicable to the largest range of design cases, we focus on the case where only a single post-dummy simulation is required, while noting that the benefits of our proposed method are multiplicative in the rare cases where multiple iterations are needed.

Although the use of Boolean operations for computation of mask layers from the "tape-out GDS" has been common practice for many technology generations, to the best of our knowledge, this work is the first report of successful use of Boolean operations to accelerate dummy pattern generation for advanced logic technologies, while simultaneously achieving improvements in layout density uniformity and a more rapid tape-out flow. In this paper, we propose a novel design and tape-out flow illustrated in Figure 2.1(b). In our proposed flow, the dummy fill operation is greatly simplified by defining the dummy fill in terms of Boolean operations, (hereafter referred to as *boolean dummy fill*). Instead of performing the dummy fill by repeatedly searching for regions of empty space in the design layout and inserting

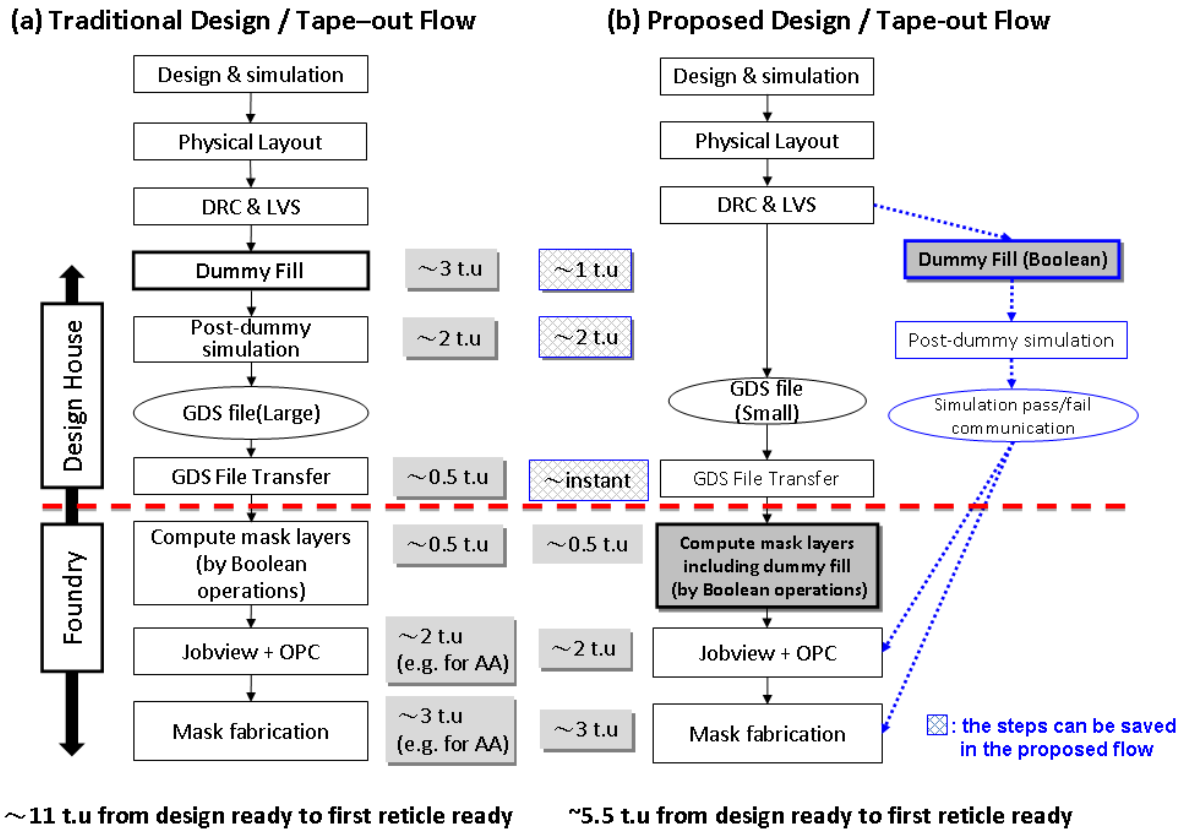


Figure 2.1: Conventional dummy fill vs. proposed dummy fill. Units of t.u. are arbitrary units of time, on the order of 24 hours, though precise execution times of course vary for different facilities.

pre-defined dummy cells, as is commonly practiced in the conventional flow, we perform the dummy fill by applying proper boolean operations similar to those which are already used to compute the mask layers based on the tape-out GDS. In other words, the dummy fill in the proposed flow is performed by the same method as the computation of mask layers by the IC foundry. Thus, the dummy generation may be merged with mask computation Booleans and may be performed by the foundry. Because the Boolean operations for dummy fill are much simpler than those typically required for mask computation, they represent only a negligible additional computational load, as foundries typically have the computational resources appropriate

to the much more complicated mask computation and OPC operations.

In advanced technology nodes beyond 65nm, most design houses perform timing checks both before and after dummy fill to insure chip performance because the dummy fill may introduce a significant parasitic capacitance load on the main circuit patterns. The parasitic capacitance change after dummy fill depends mainly on two factors: (1) dummy cell geometry, and (2) the distance between the dummy cells and the main circuit pattern. As a result, dummy fill operations have the potential to affect chip timing and this effect becomes more significant in more advanced technology nodes because more and increasingly complex dummy patterns are required for the optimization of process variation. Generally, the performance impact of the dummy patterns results in a simple performance shift, but does not result in changes to either design layout or dummy fill. The post-dummy simulation is thus generally a necessary safety check to insure that the design delivers adequate performance, but rarely drives changes in the database. However, for demanding high-performance designs, in principle, it is possible that the results of post-dummy simulations may result in one or more iterations of either design layout change or fine-tuning of the dummy generation to compensate for timing impact. Our proposed method would remain beneficial even in such cases. The main benefit of our method is due to (a) the database retaining a high degree of hierarchy when the dummy fill is performed, and (b), the use of Boolean operations rather than an iterative algorithm to perform the dummy fill. Our proposed method permits fine-tuning of dummy cell geometry to the same degree as conventional dummy fill, while still retaining the benefits of improved hierarchy and speed of execution. If additional post dummy simulations or iterations on dummy fill geometry are required, our proposed method provides an even greater net gain because each iteration can be performed more quickly on a smaller hierarchical database.

Further efficiency can be obtained because the Boolean dummy fill and post-dummy simulations to assess the effect of dummy patterns on product performance, not only incur a much reduced computational load by the use of boolean dummy fill, but also these tasks can be performed simultaneously with the computation of the mask layers and OPC by the foundry, so several days of overhead can be eliminated. Once post-dummy simulation is complete, approval for mask fabrication (or mask release) can be communicated and mask fabrication (or wafer processing) can proceed immediately. In principle, the design house could completely forgo dummy fill and post-simulation, but even when it is desired, it imposes no delay in the tape-out flow. Although the work of Ref. [23] demonstrated a dummy-insertion flow with some features which are present in our proposed flow, and may have been implemented by Boolean operations, there was no detailed discussion of the method of implementation, performance metrics, or the use of this flow to accelerate the overall tape-out process, and the scope was limited to adding dummy poly to address a specific process issue in a relatively old technology having poly spacing of $\sim 0.35\mu\text{m}$. Nevertheless, because the work of Ref. [23] is the only previous report of a possible partial implementation of our proposed flow of which we are aware, similarities and differences with this work will be specifically addressed in the relevant sections below.

In summary, boolean dummy fill and our proposed design and tape-out flow provides and enables the following benefits. First, the runtime overhead of the proposed dummy fill depends on the runtime of performing additional boolean mask operations among the GDS layers, which is much faster than performing extensive GDS searching operations as is done in conventional dummy fill algorithms, and thus is a smaller computational load for the design house. Second, boolean dummy fill delivers improved pattern uniformity because the pattern uniformity of the boolean

dummy fill is not limited by the dimensions of the predefined dummy cells as is the case with conventional dummy fill algorithms. Third, dummy generation and post-simulation at the design house are performed in parallel with dummy generation, mask layer computation and OPC (and potentially even mask writing), eliminating several days of overhead from the tape-out flow, enabling more rapid silicon verification of new or modified designs. The experimental results below, based on advanced process technologies will further demonstrate the efficiency and effectiveness of the proposed design and tape-out flow enabled by boolean dummy fill.

2.1 Background

2.1.1 Long-range vs. Short-range Dummy Patterns

Dummy patterns can be categorized as either *short-range* or *long-range* dummy patterns. Short-range dummy patterns are placed close to the active patterns to eliminate potential local process variation, such as mechanical-stress effects [18] and proximity effects [19] [20]. The mechanical-stress effect is caused by the lattice mismatch of different films. The level of strain of different active regions may affect the silicon bandgap, diffusivity of impurities, and mobility, and thus may result in variation of devices' threshold voltage, transconductance, saturation drain current, and off-state drain current, even though the critical dimensions (CD) of the devices are the same [18]. Dummy patterns can be inserted as stress buffers to balance this mechanical stress. The optical proximity effect is generated by optical diffraction during lithography such that the CD of an isolated layout pattern may be different from that of a nominally identical pattern in a dense layout. To resolve this proximity effect, dummy patterns with comparable dimensions and regular spacing must be inserted in the immediate neighborhood of isolated active patterns. Thus, the dimensions of short-range dummy patterns are relatively small [19] [20].

On the other hand, long-range dummy patterns are placed at a greater distance from the active patterns to eliminate potential global process variations caused by unbalanced layout density and low layout uniformity over distances of several microns or tens of microns. The sources of such longer range process variation include thermal anneal (RTA, Flash, and Laser) [21], CMP [9] [22], and etch micro-loading [9] [24]. Since the length scale associated with these process variations is larger, the dimensions of long-range dummy pattern cells are relatively large.

2.1.2 Dummy-Pattern Style

Figure 2.2 shows different layout styles of dummy patterns. The first style (also the most conventional one) is the grid style, in which dummy patterns are placed in identical rectangles and aligned in the Y-direction with a fixed X-direction spacing. The second style is the staggered style, in which rectangular patterns in adjacent columns are placed with a fixed offset in the Y-direction. The staggered style results in more uniform fluid dynamics and reduces the competition for etch reactants to achieve a more uniform etch rate, and thus this style is preferred for reducing micro-loading effects [9]. The third style is the diamond-shape style, in which the dummy patterns are identical diamond shapes and placed with fixed spacing. This diamond-shape style is most commonly used for dummy metals because it minimizes the effective coupling capacitance [10]. The choice of the dummy-pattern style is determined by the characteristics of the adopted process technology and may differ for different IC foundries.

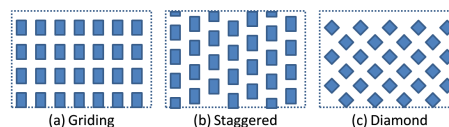


Figure 2.2: **Example of different dummy-pattern styles.**

Note that the example shown in Figure 2.2 is only for one GDS layer. In practice, dummy patterns are simultaneously added into different layers, such as poly, diffusion, and metal. Thus, dummy patterns are required for each corresponding layer. In fact, the dummy patterns for both poly and diffusion layers are designed and inserted simultaneously. The dummy pattern for metal layers are designed and inserted independently from the poly and diffusion layers.

2.1.3 Computation of Mask Layers Using Boolean Operations

Boolean operations have been widely used for IC mask generation [34] [35]. When the GDS file of a design is sent to an IC foundry or mask vendor, all the mask layers must first be computed by Boolean operations from the tape-out GDS layers (often referred to as the CAD layers). Note that the number of GDS layers present in a typical tape-out GDS file is around 20, whereas the number of mask layers in a typical CMOS mask set is around 40 (depending on the specific process technology). Some mask layers, such as PW, Ldd, and SiGe, are not present in the tape-out GDS, but can be computed by applying boolean mask operations (such as NOT, OR, SIZING) on the tape-out GDS layers. In general, the runtime of computing a mask layer is less than 1 hour. This runtime is only a small portion of the entire process of mask-set generation, whose computation time is dominated by optical proximity correction (OPC). To perform OPC on one mask layer may take days for large advanced designs.

2.2 Boolean-Operation Dummy Fill

2.2.1 Overall Flow

According to the design flow shown in Figure 2.1(b), the proposed boolean dummy fill is performed at the same time as the mask layers are computed from

the GDS layers. In other words, with boolean dummy fill, it is no longer necessary to search for empty space in the design layout and insert dummy cells as in the conventional flow. Instead proper boolean mask operations must be designed for directly generating the corresponding mask layers with dummy patterns inserted. In principle, the responsibility for performing dummy fill can be transferred from design houses to IC foundries, where it is performed much more efficiently as part of the mask layer boolean operations. Design houses need only provide their verified design layout without the addition of dummy patterns, and the IC foundries can proceed as suggested in the flow of Figure 2.1(b) . If post-simulation is required, the design house can perform the small subset of the mask boolean operations which perform the dummy fill, and carry out post-dummy simulation while the foundry is performing the OPC computations for the first layers in the process flow. This achieves the following benefits. First, less computation time is required at the design house due to the improved computational efficiency of boolean dummy generation. Secondly, the computation time for dummy generation and post-dummy simulation at the design house is no longer a bottle-neck in the tape-out flow due to the parallel generation of dummy patterns at the design house and foundry. Thirdly, the size of the tape-out GDS file transmitted from the design house to the foundry is dramatically reduced, so this transmission delay becomes negligible. Since the runtime of computing the additional boolean operations for dummy fill is far less than that of the original boolean operations used to compute the mask layers, there is relatively little additional computational overhead at the foundry.

The following are the input and output of the proposed boolean-operation dummy fill.

Input:

- Active patterns in the tape-out GDS file provided by the design house.
- The dummy-pattern style of both long-range and short-range dummy patterns.
- The spacing between the active patterns and dummy patterns.
- The placement range (distance to the active pattern) of short-range and long-range dummy patterns.

Output:

- A set of boolean mask operations which can generate the mask layers with dummy patterns inserted.

2.2.2 Detailed Steps of Boolean-operation Dummy Fill

Figure 2.3 lists the six steps used in the proposed boolean-operation dummy fill, including (1) dummy-tile spreading, (2) identifying active-pattern regions, (3) generating non-overlapping dummy patterns, (4) jog removal, (5) small island removal, and (6) combining active patterns and dummy patterns. The details of each step are presented in the following subsections. Steps 1-4 and 6 were performed by a single Mentor Calibre script. For reasons of convenience, Step 5 was performed by a separate script as will be explained below, but the entire flow could easily be performed by a single integrated script, which would result in run-time reductions even greater than those we report here. GDS file format is used throughout the flow. Note that in the following example, we generate dummy patterns only on the poly layer and use only the short-range dummy cells. A similar method can be applied to generate dummy patterns for other layers. We will later show how to generate dummy patterns with the use of both short-range and long-range dummy cells simultaneously. A similar flow for dummy poly insertion is described by Ref.

[23] in a successful addition of dummy poly patterns to reduce the difference in contact ILD deposition and etch behaviors between gates with neighboring gates on both sides and gates with no neighboring gate on one side. Our flow and the flow of Ref. [23] both exploit (1) dummy-tile spreading, (3) generating non-overlapping dummy patterns, (4) jog removal, (5) small island removal, and (6) combining active patterns and dummy patterns, and these steps may have been implemented using Boolean operations, but Ref. [23] does not state the software tool (e.g. Mentor Calibre) or algorithm (e.g. boolean operation or search algorithm) used to implement the flow in their work. Because the flow of Ref. [23] begins with identifying the poly edge candidates for dummy pattern addition prior to dummy tile spreading, it appears probable that a search algorithm was employed in this work for this initial step. Further, the algorithm implemented by Ref. [23] appears to be insufficiently robust for fully automated use, because the flow of Ref. [23] explicitly includes a final step where the layout after dummy insertion must again be searched for non-compliant poly edges, and these must be edited manually. This final search of the database after dummy insertion is likely to incur substantial computational overhead. Finally, Ref. [23] was implemented on a relatively old technology (having poly pitch of $\sim 0.35\mu\text{m}$), and the scope appears to be limited to the addition of dummy poly patterns to address a specific process issue, without a discussion of the computational overhead or optimization of the implementation algorithm. Our work is broadly applicable to multiple sizes of dummy patterns on multiple layers, has been successfully used in a fully automated mode on multiple customer products, and here-in we will show that implementation by Boolean operation achieves superior computational performance, improved pattern uniformity, and accelerated tape-out compared to the current state of the art dummy insertion tool provided by a major foundry.

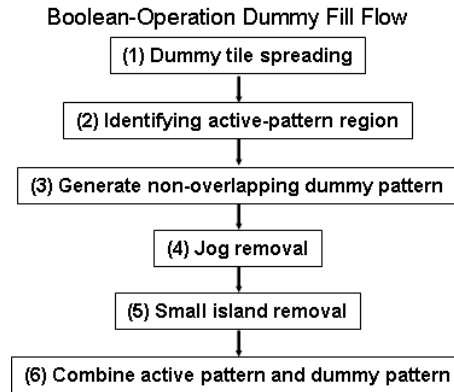


Figure 2.3: Steps of the proposed boolean-operation dummy fill.

2.2.2.1 Dummy-tile Spreading

The objective of the dummy-tile spreading is to generate the dummy patterns (short-range or long-range) for the entire poly, diffusion, or metal layer. Those full-layer dummy patterns are then used as the operands of the boolean mask operations to compute the final physical dummy patterns. This full-layer dummy pattern is generated by duplicating the predefined dummy tile over the entire design window. The predefined dummy tile is designed according to the adopted dummy-pattern style. Figure 2.4 shows examples of designing the corresponding dummy tile for dummy-tile spreading. Once the dummy tile is defined, this layout spreading can be easily performed by a commercial layout tool, such as Virtuoso, Laker, or Mentor Calibre, as was used in this work.

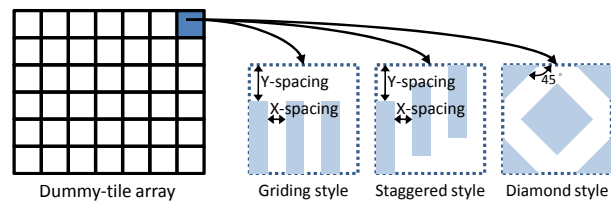


Figure 2.4: Dummy Tile Design.

2.2.2.2 Identifying Active-Pattern Region

Before generating the final dummy patterns, it is first necessary to identify the active-pattern region and eliminate this region from the full-layer dummy pattern. This active-pattern region includes the area of the active patterns as well as the spacing between the active patterns and dummy patterns, which can be computed by Equation 3.1. Note that when calculating the active-pattern region for dummy poly patterns, the active patterns should include both poly and diffusion patterns since a poly dummy line cannot overlap with active diffusion patterns. The active metal patterns are excluded from the above active-pattern region. In the following discussion, GDS layer IDs (PO;0) and (PO;1) represent the active patterns and the dummy patterns on the poly layer, respectively. (AA;0) and (AA;1) represent the active patterns and the dummy patterns on the diffusion layer, respectively.

$$(AA;0 \text{ OR } PO;0) \text{ SIZING } s_1 \quad (2.1)$$

In Equation 3.1, (PO;0 OR AA;0) represents the union of active poly patterns and active diffusion patterns. The (SIZING s_1) operation enlarges the designated patterns by $s_1 \mu m$, where s_1 denotes the minimum spacing between active patterns and dummy patterns.

2.2.2.3 Generating Non-Overlapping Dummy Pattern

In this step, we compute the final dummy patterns, denoted as DP , by eliminating the active-pattern region from the full-layer dummy pattern. Equation 2.2 lists the corresponding boolean operations.

$$DP = PO;1 \text{ NOT } ((AA;0 \text{ OR } PO;0) \text{ SIZING } s_1) \quad (2.2)$$

Note that the operation "NOT" represents the difference operation. The output of the operation (A NOT B) is the layout patterns which are in A but do not overlap with the patterns in B.

2.2.2.4 Jog Removal

After eliminating the active-pattern region from the full-layer dummy patterns, some jog patterns or small-island patterns may exist among the dummy patterns as shown in Figure 2.5(a). The jog patterns and small-island patterns could become a source of defectivity during wafer processing, resulting in degraded yield [27]. In addition, these small-island patterns also significantly increase the computational overhead of the optical proximity correction due to their irregular shapes.

The objective of the current step is to remove the jog patterns from the dummy patterns by using the following boolean operations. First, Equation 2.3 is used to shrink the dummy patterns by $x \mu m$, where $x = \frac{w}{2} - u$, w represents the width of the adopted dummy pattern, and u represents the minimum layout unit. The minus sign used in the SIZING operation denotes shrinking the feature size. As illustrated in Figure 2.5(b), after performing Equation 2.3, the jog patterns completely disappear and the original regular patterns have been reduced to the minimum permitted dummy feature width.

$$\text{DP SIZING } -x \tag{2.3}$$

Next, Equation 2.4 is used to enlarge the shrunk patterns by $x \mu m$. Since the jog patterns have already been eliminated, Equation 2.4 simply restores the regular

patterns back to their original size as illustrated in Figure 2.5(c), and hence obtains dummy patterns without jogs (denoted as DP_JR).

$$DP_JR = (DP \text{ SIZING } -x) \text{ SIZING } x \quad (2.4)$$

In other words, by applying both Equation 2.3 and Equation 2.4, any pattern with a width less than w will be eliminated and the other patterns are unchanged

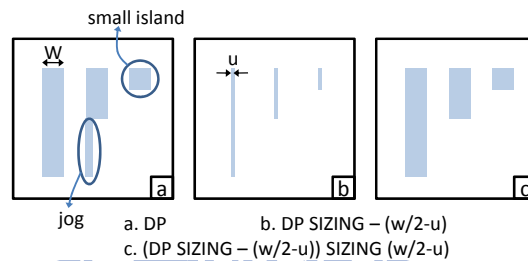


Figure 2.5: **An example of the jog removal.**

2.2.2.5 Small-Island Removal

The next step focuses on removing the small-island patterns, i.e. the patterns whose area is smaller than the minimum area permitted by the design rules of the particular process technology. The previously described jog removal can only eliminate the patterns whose width is smaller than a predefined constraint, but in many process technologies a shape with dimensions exceeding the minimum permitted width may still be forbidden by a minimum area design rule. Thus, small-island removal is performed by a Mentor Calibre DRC (design rule checking) script, which identifies and removes all such small island patterns. This step was performed by a DRC script which was separate from the main Calibre script used for the other Boolean operations only because such a DRC script was conveniently available. Performing this step with a separate script caused additional runtime overhead because

the GDS output of the jog removal operation was flattened for input into the small island removal script, but its runtime was still much faster than conventional dummy insertion as shown in Figure 2.1.

2.2.2.6 Combining Active Patterns and Dummy Patterns

The final step is to combine the generated dummy patterns with the active patterns (PO;0) by using the boolean operation *OR*. Equation 2.5 shows the boolean operations combining all the above steps, which can generate the final poly layer combining the dummy patterns and the active patterns.

$$\begin{aligned}
 & DP_JR \text{ OR } (AA;0 \text{ OR } PO;0) \\
 = & (((PO;1 \text{ NOT } ((AA;0 \text{ OR } PO;0) \text{ SIZING } s_1)) \\
 & \text{SIZING } -x) \text{ SIZING } x) \text{ OR } (AA;0 \text{ OR } PO;0) \tag{2.5}
 \end{aligned}$$

2.2.2.7 Example

Figure 2.6 shows an example of applying the proposed boolean-operation dummy fill with only short-range dummy patterns. In this example, the result and the boolean operations for each step are shown the sub-figures. The active pattern used in this example is an inverter.

2.2.3 Dummy Fill with Long- & Short-range Dummy Patterns

In the following section, the use of boolean-operation dummy fill to simultaneously place both long-range and short-range dummy patterns is demonstrated. Again, the example used is the case of generating dummy poly, but the dummy patterns for other layers can be obtained in a similar manner. In the following discus-

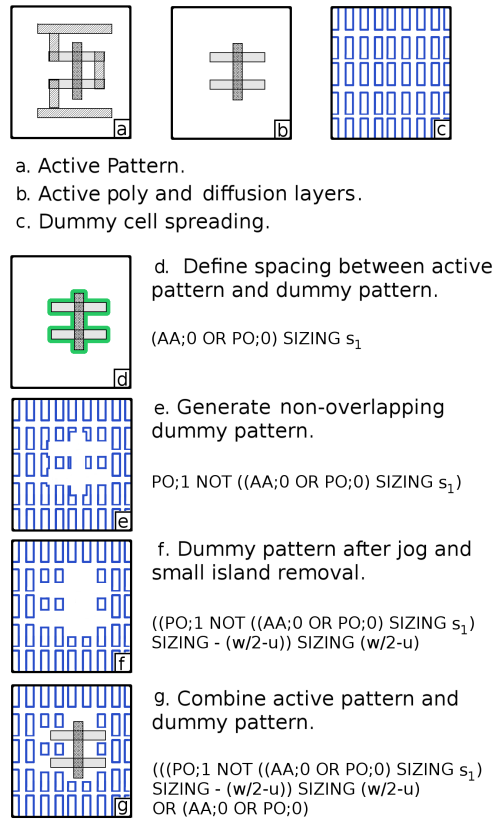


Figure 2.6: **Example of boolean-operation dummy fill using only short-range dummy patterns.**

sion, $(PO;1)$ and $(PO;2)$ represent the full-layer dummy patterns after dummy-tile spreading for the short-range dummy pattern and the long-range dummy pattern, respectively. The dummy generation is determined by the following parameters: s_1 represents the minimum spacing between active pattern and short-range dummy patterns, s_2 represents the farthest distance outside the active-pattern region where a short-range dummy pattern can be placed, and s_3 represents the minimum spacing between the short-range patterns and the long-range patterns. These three parameters are determined by the selected process technology and predefined as inputs of the boolean dummy fill

The boolean dummy fill including both long-range and short-range dummy patterns, illustrated in Figure 7, follows the same steps shown in Figure 2.3 but the step of generating non-overlapping dummy patterns is slightly more complicated than that described in Section 2.2.2.3. Short-range dummy patterns can only be placed within a distance s_1 and a distance $(s_1 + s_2)$ of the active patterns. Thus, to compute the short-range dummy patterns, the short-range dummy patterns excluding the active-pattern region are generated by using Equation 2.6, where the $((AA:0 \text{ OR } PO;0) \text{ SIZING } s_1)$ represents the active-pattern region. The resulting patterns are denoted as SRD_in and illustrated in Figure 2.7(c).

$$SRD_in = PO;1 \text{ NOT } ((AA:0 \text{ OR } PO;0) \text{ SIZING } s_1) \quad (2.6)$$

Of course a simple modification of Equation 2.6 permits different values of s_1 for AA:0 and PO;0 layers if required, but generally the same value can be used because the spacing of these two layers follows a fixed relationship set by the technology design rules. Second, the short-range dummy patterns placed outside the feasible range of the short-range patterns are generated by Equation 2.7, where $((AA:0 \text{ OR } PO;0) \text{ SIZING } (s_1 + s_2))$ forms the largest region within which short-range dummy patterns can be placed around the active patterns. This pattern, denoted SRD_out is illustrated in Figure 2.7(d).

$$\begin{aligned} & SRD_out \\ = & PO;1 \text{ NOT } ((AA:0 \text{ OR } PO;0) \text{ SIZING } (s_1 + s_2)) \end{aligned} \quad (2.7)$$

From the above intermediate results, the final short-range dummy patterns,

denoted as SRD , are calculated by eliminating SRD_{out} from SRD_{in} as shown in Equation 2.8. Figure 2.7(e) illustrates the resulting SRD .

$$SRD = SRD_{in} \text{ NOT } SRD_{out} \quad (2.8)$$

Next, the long-range dummy patterns need to be placed outside the region which is at a distance s_3 from the short-range patterns, i.e., $(s_1 + s_2 + s_3)$ a distance from the active patterns. Thus, the long-range dummy patterns, denoted as LRD , are obtained by eliminating the invalid region from the full-layer long-range dummy patterns as shown in Equation 2.9. Figure 2.7(f) illustrates the resulting LRD .

$$LRD = PO;2 \text{ NOT } ((AA;0 \text{ OR } PO;0) \text{ SIZING } (s_1 + s_2 + s_3)) \quad (2.9)$$

Finally, the final poly dummy patterns, denoted as FPD , can be computed by combining SDR and LRD as shown in Equation 2.10. Figure 2.7(g) illustrates the final poly dummy patterns.

$$FPD = SDR \text{ OR } LRD \quad (2.10)$$

Figure 2.7(h) illustrates the final poly layer including both dummy and active patterns.

Note that the above discussion focused primarily on the steps required for generating non-overlapping dummy patterns. The boolean operations used for jog

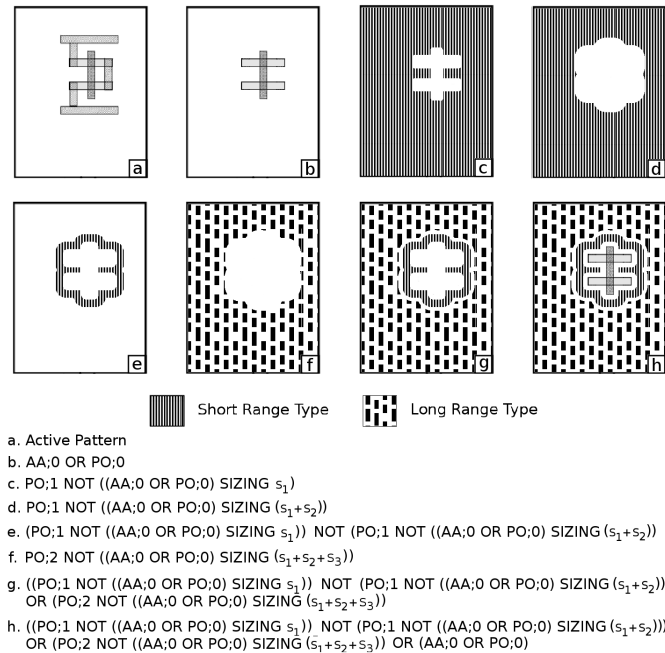


Figure 2.7: **Example of boolean-operation dummy fill using both short-range and long-range dummy patterns.**

removal were omitted in order to simplifying the illustration. In practice, the complete boolean operations required for dummy fill with both short-range and long-range dummy patterns are actually more than those listed in the above equations and Figure 2.7.

2.2.4 Dummy Cell Initial Definition For Optimizing Layout Density Uniformity

In the above discussion, boolean dummy insertion has been described for the case where a single set of short-range and long-range dummy cells is utilized throughout the chip, following the standard practice for conventional dummy pattern insertion. It will be shown by our experimental results that even with this constraint, boolean dummy insertion delivers improved cross-chip pattern density uniformity. However, because boolean dummy insertion is performed by simple chip-

level boolean operations, further optimization of layout density uniformity can be achieved by the appropriate initial definition of different dummy cells for different regions of the chip based on an initial survey of the layout density of the tape-out GDS. Before beginning the boolean dummy insertion flow, the tape-out GDS is surveyed, and the layout density is calculated for each rectangular sub-section of the chip as shown in Figure 2.8, where the size of the sub-section is determined by process technology requirements. Then, short-range (and if necessary, long-range) dummy cells can be initially defined with layout densities determined by the initial layout density of each sub-section in the tape-out GDS. These cells can either be defined to attempt to achieve a target layout density, or they may simply be adjusted to tighten the statistical distribution of the layout density. In either case, the easy application of boolean dummy insertion, utilizing dummy cells based on the initial layout density, is certain to improve layout density uniformity even further than the improvements which are demonstrated by our experimental results. Of course the same concept can be applied to pre-define dummy cells for conventional search-based dummy fill algorithms, but, as noted above, the improved computational efficiency of boolean dummy fill enables this technique to be applied more easily.

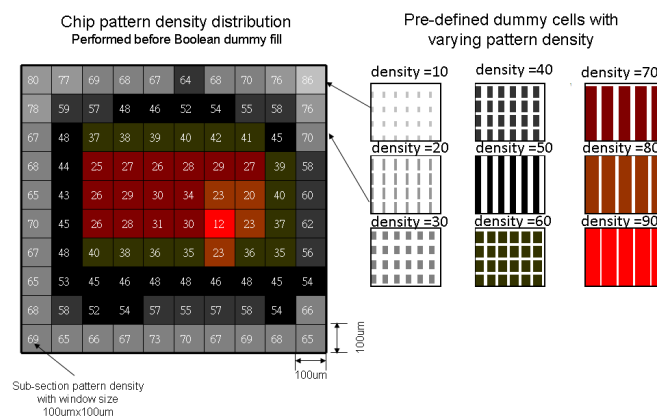


Figure 2.8: Definition of dummy cells for different chip sub-regions based on initial layout density .

2.3 Experimental Results

2.3.1 Conventional Dummy Fill & Test Cases

In our experiments, we compare the proposed boolean-operation dummy fill with a conventional dummy-fill flow, which has been applied to several product lines and is provided to customers by a well-known IC foundry. The concept of this conventional dummy fill is to insert the predefined dummy cells, which are also provided by the IC foundry, into the empty space of the design layout based on an insertion algorithm. Each predefined dummy cell has different dimensions as well as different insertion rules. Also, the predefined dummy cells can be designed with multiple-layer patterns and are sorted by their cell size. When empty space is found, the conventional dummy fill algorithm will always try to insert the largest dummy cell possible according to the cell's insertion rules. Note that the final GDS-file size is directly proportional to the number of dummy cells inserted. Thus, by inserting the largest possible dummy cell, the GDS-file size can be minimized. The conventional dummy fill is implemented by a Calibre script, and has been extensively optimized under the constraints imposed by the need to maintain a reasonable file size and run time. The test cases in our experiments are various representative products manufactured by the same IC foundry. The test cases include a 16Mb SRAM design, a mixed-signal design, and two logic designs developed with an advanced process technology. All the reported results were computed on a PC with a 2.8GHz Opteron CPU (4 cores) and 64GB main memory.

2.3.2 Comparison of Runtime and GDS File Size

Table 2.1 shows the results for applying the conventional dummy fill to each test case. In Table 2.1, Column 2 lists the chip size, columns 3 and 4 list the GDS-file size before and after applying the conventional dummy fill, respectively,

and column 5 lists the ratio of these two file sizes. Column 6 lists the runtime for applying conventional dummy fill. As the results show, the GDS-file size may increase dramatically (29.1X in average) after the conventional dummy fill. This ratio ranges from 13.2X to 56.5X for different test cases and depends on the layout density of the original design, which in turn affects the number of dummy cells to be inserted. File sizes for the largest test case before and after the conventional dummy fill are 360MB and 8.4GB, respectively. The runtime of the conventional dummy fill is determined by the chip size and the number of inserted dummy cells. The longest runtime of the examples presented here was slightly more than 5 hours.

test case	chip size (μm^2)	GDS file size (byte)		ratio b/a	runtime (sec.)
		before fill (a)	after fill (b)		
Memory	3800x4000	29,818,306	1,684,161,736	56.5x	11619
Mix_signal	2200x660	191,690,752	2,526,404,608	13.2x	3653
Logic1	2200x6240	39,909,376	930,647,608	23.3x	18003
Logic2	3300x3300	360,988,072	8,359,665,856	23.2x	4207
average				29.1x	9371

Table 2.1: **Resulting GDS file size and runtime for conventional dummy fill.**

Note that the test cases used in this experiment are substantially smaller than typical present-day advanced designs. However, the resulting GDS-file size already exceeds the memory limitation of an average PC. In fact, we attempted to run this experiment by reducing the main memory of the PC from 64GB to 16GB, but these experiments failed to run due to insufficient memory. This result demonstrates the large computational overhead of conventional dummy fill. Also, it can be expected that the GDS-file size increase by the conventional dummy fill may become even worse in the near future when the need of short-range patterns further increases for more advanced processes and thus a larger number of smaller dummy cells must be inserted. This experimental result further demonstrates the advantage of the proposed boolean-operation dummy fill and eliminates the need for the tape-out GDS to contain dummy fill patterns.

Next, Table 2.2 shows the runtime of performing the proposed boolean-operation dummy fill. In Table 2.2, Column 2 lists the runtime of performing the boolean operations only for computing original mask layers (without dummy fill). Column 3 lists the extra runtime of performing the boolean operations for generating the mask layers including dummy fill, i.e. steps 1-4, and step 6 of Figure 2.3. Column 4 lists the runtime for applying the small-island removal Calibre script (step 5 of Figure 2.3). Although in the implementation reported in this work, for convenience of implementation, the small island removal step was implemented with a separate, readily available Calibre DRC script, which was different from the script used for the remainder of the steps in Figure 2.3, combining all operations in Figure 2.3 into a single Calibre script is a straightforward programming exercise. Because much of the processing time in Column 4 is due to the disruption of the GDS hierarchy for input into the DRC script, a single integrated script would have much lower execution time for this operation. Figure 2.9 illustrates the disruption of the GDS hierarchy of an array of dummy patterns after applying one boolean operation.

The overall runtime overhead of the proposed dummy fill is listed in Column 5, which is the summation of Column 3 and Column 4. The average runtime overhead of the proposed dummy fill is 445 seconds, which is 1/21 of that of the conventional flow (9371 seconds). This result demonstrates the efficiency of performing dummy fill based on boolean mask operations. In this case, the same computing platform was used for both conventional dummy fill and the proposed dummy fill. In practice, the computers typically used for mask set computation and OPC are much more powerful than the workstations generally used in design houses. Thus, the actual runtime overhead of the proposed dummy fill would be even shorter if performed on a computing platform typically utilized for OPC. In addition, the

additional runtime overhead of the proposed dummy fill, on average, is around 1/4 of that of computing the original mask layers, which is also a small portion of the entire process of mask-set generation when the long runtime of OPC is considered.

The different amounts of overhead for these respective designs can be qualitatively explained by the percentage of die area devoted to memory, logic, and mixed-signal I.P. respectively. For the "Memory" test case, most of the die area consisted of high density memory, so relatively few dummy patterns needed to be added. As a result, the non-overlapping dummy pattern area (generated by step 3 of Figure 2.3) is quite small, and the remaining steps in the flow consume relatively little computation time. On the other extreme, for the test case "Logic 2", a relatively large portion of the chip area consisted of random logic with a layout style which was not particularly dense. As a result, a relatively large number of dummy patterns were added. Note however that the small-island removal time for this case is comparable to the memory case, because the low density layout style required less use of small dummy cells, and thus less small island removal. The test case "Mix_signal" had a relatively high percentage of mixed-signal and/or RF I.P., and for this I.P. the dummy patterns had been drawn in the design GDS to optimize matching and to meet electrical requirements for some of the dummy patterns (e.g. requirements that some dummy patterns be grounded, and others be connected to power planes). Thus, similar to the "Memory" test case, there was a relatively small area that required dummy fill, resulting in a small non-overlapping dummy pattern area, and a low small-island removal time. Test case "Logic 1" utilized a layout style of much higher density than test case "Logic 2", so while the amount of dummy pattern area required was not as large as "Logic 1", a larger number of small dummy patterns were required, resulting in a larger overhead for small-island removal. Only in the case of "Logic 2" was the overhead more than 50% of the computation time

required for mask layer generation.

Table III shows the file sizes obtained by applying the proposed dummy fill to each test case. Following the convention of Table I, in Table III, column 2 lists the chip size, and columns 3 and 4 list the GDS-file size before and after applying the proposed dummy fill. The ratio of the file size after dummy fill to before dummy fill is below 5X, which is a relatively small rate of file size increase compared with conventional dummy fill. This small rate of increase can be obtained because the GDSII file after the proposed dummy fill still maintains a certain degree of hierarchy, resulting in much less file size increase compared to conventional dummy fill which does not take advantage of database hierarchy.

Due to increasing design complexity, in recent years, an increasing portion of the design community is adopting OASIS file format rather than the traditional GDSII. Typically, transitioning from GDSII to OASIS reduces database file size by a factor of roughly 7-10X, where the exact value is a function of the degree of cell hierarchy. Based on experience in advanced technology tape-outs which submit databases in OASIS format, it has been observed that the size ratio between GDSII and OASIS remains nearly constant before and after dummy fill, i.e. approximately 7-10X. The benefits we describe in this paper exist no matter which of these two file formats is used. However, the precise size ratio between the two formats for a given design depends on the degree of cell hierarchy employed.

test case	(a)original mask layers operation	proposed flow			(% overhead (d/a))
		(b)dummy fill operations steps 1-4 and 6	(c)small-island removal	(d)total =b+c	
Memory	8265	275	68	343	4.2%
Mix_signal	825	95	59	154	18.7%
Logic1	4965	180	540	720	14.5%
Logic2	945	450	113	563	59.6%
average	3750	250	195	445	24.3%

Table 2.2: **Runtime of the proposed boolean-operation dummy fill.**

test case	chip size (μm^2)	GDS file size (byte)		ratio b/a
		before fill (a)	after fill (b)	
Memory	3800x4000	29,818,306	145,272,352	4.84x
Mix_signal	2200x660	191,690,752	876,345,355	4.57x
Logic1	2200x6240	39,909,376	91,074,334	2.28x
Logic2	3300x3300	360,988,072	424,575,146	1.18x
average				3.22x

Table 2.3: Resulting GDS file size and runtime for the proposed boolean-operation dummy fill.

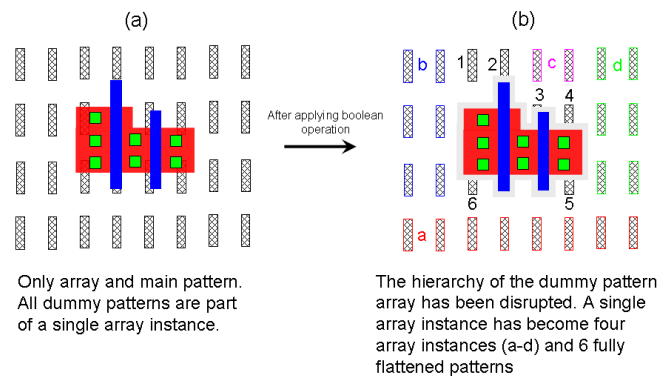


Figure 2.9: Illustration of disruption of the GDS hierarchy of a dummy pattern array after applying one boolean operation

2.3.3 Chip Layout after Dummy Fill

Figure 2.10 shows the layout of a sub-block in the test case "Logic1" after the boolean-operation dummy fill is applied, where both the short-range and long-range dummy patterns can be clearly observed. Figure 2.11(a) and Figure 2.11(b) show the layout of the full chip and the layout of a selected region for the SRAM test case, respectively. Figure 2.11(c) and Figure 2.11(d) show the layouts of the selected region after applying the conventional dummy fill and the boolean-operation dummy fill, respectively. The uniformity of the dummy patterns in Figure 2.11(d) is higher than that in Figure 2.11(c). This is because the dummy patterns are inserted based on the unit of a dummy cell in the conventional flow and the empty space remaining in the layout may not be large enough to place the smallest dummy cell. This problem can be solved by providing smaller predefined dummy cells. However, this

would almost certainly increase the runtime and GDS file-size of the conventional flow because larger numbers of smaller dummy cells would need to be inserted. On the other hand, the dummy patterns in the proposed flow are generated by eliminating the active-pattern region from a full-layer dummy pattern, so empty space is eliminated more effectively than in the case of the conventional flow. This result demonstrates the high pattern uniformity achieved by the proposed boolean-operation dummy fill.

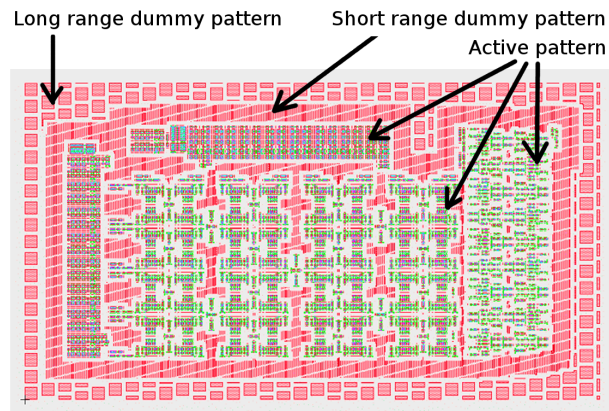


Figure 2.10: A portion of the layout of the test case Logic1 after performing the boolean-operation dummy fill.

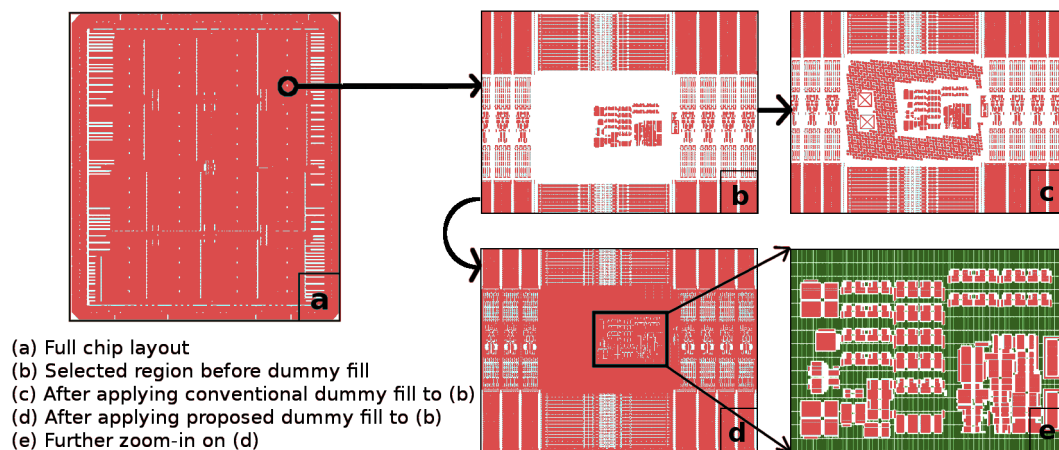


Figure 2.11: Layout of the test case "Memory" after applying conventional dummy fill and boolean-operation dummy fill respectively.

2.3.4 Pattern Density after Dummy Fill

The pattern density of a real product chip after performing dummy fill was extracted by the Mentor Calibre tool with a window size of 20X20um. Figure 2.12 compares the surface and contour plots for the pattern density of the AA layer after performing both traditional and boolean dummy fill. As shown, there are three major areas where active patterns were originally present in the design. In the case of traditional dummy fill, many large changes in layout density occur in the immediate neighborhood of the active patterns. These density changes are caused by insufficient open area for insertion of pre-defined dummy cells. However, these changes in pattern density can be significantly reduced by the proposed boolean dummy fill methodology, as shown in left plot in same Figure. This improvement is primarily due to the boolean "NOT" boolean following the pre-defined dummy cell spreading. Dummy insertion is performed by boolean dummy fill as long as the open area is larger than S1 and the dummy features pass small island removal as described in section 2.2.2. In other words, the proposed boolean dummy fill algorithm can improve the chip pattern density uniformity, especially on areas very close to active patterns, which is very difficult to achieve using traditional dummy insertion algorithms. It is possible that comparable pattern density uniformity could be achieved by the traditional algorithm by a "brute force" approach of adding additional smaller pre-defined dummy cells, but such an approach would greatly increase the algorithm run-time as well as the file size after dummy insertion. We further note that in this example, pattern density uniformity was improved even when the same set of short and long-range dummy cells was utilized for the entire chip. Further improvement is clearly possible based on an appropriate initial definition of different dummy cells for different sub-regions of the chip where the density of the dummy cells is derived from the initial layout density, as described previously.

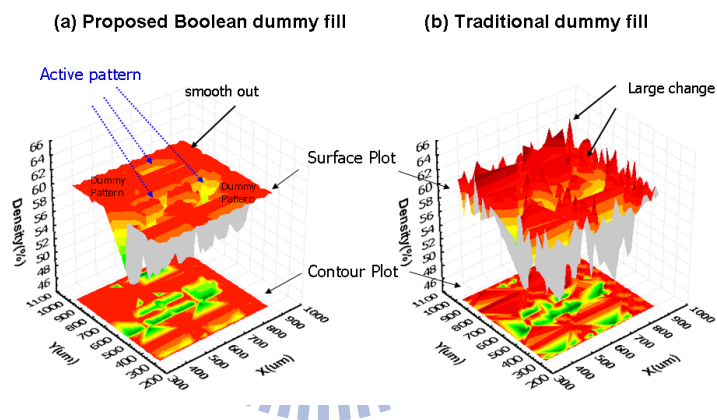


Figure 2.12: Surface and contour plot of pattern density after performing the traditional and proposed boolean dummy fill.

Chapter 3

Mask versus Schematic – An Enhanced Design-Verification Flow for First Silicon Success

Layout-versus-schematic (LVS) verification software extracts a netlist from the GDS file containing a circuit's physical layout and identifies all discrepancies between this netlist and the original design netlist created prior to the GDS layout. Both connectivity and device attributes (device type e.g. MOS, resistor, capacitor, device width, device length, etc.) are checked for equivalence. It has been used successfully on countless chip designs for several years [28][29]. Some algorithm optimization was proposed and implemented in QA LVS rule deck. By using this technique, users can quickly obtain an error free LVS rule deck [30]. More recently in advanced CMOS technologies, the LVS tool is also used to extract the parameters describing the surrounding layout of each transistor to assess the impact of Layout Dependent Effects [LDE] such as Well Proximity Effect (WPE), Active-Spacing Effect (OSE) and Poly Spacing Effect (PSE)... etc. [32].

However, in advanced process technologies, the photomask layout is no longer an exact replica or minor modification of the design layout. The data used for input to the LVS performed at the design stage are not the actual final on-masks patterns which will be fabricated in silicon. Some operations performed after the LVS verification step, such as dummy fill operations, mask boolean operations to generate non-drawn layers, and OPC (optical proximity correction), may substantially

change the layout environment near a particular device. A detailed comparison of these three types of GDS operations is illustrated in Fig. 3.1. The dummy fill operations are performed by placing the pre-defined dummy cells into the empty space of a design layout to balance its feature density, which will not change the geometry of the active pattern [36][37][38][39]. In general, a verification step to check and quantify the impact of the generated dummy pattern on the original drawn active pattern should be executed once the dummy fill operation is completed to ensure that there is no negative impact resulting from the dummy fill. OPC techniques are commonly employed to improve the photo process margin by either modifying the initially drawn features or by introducing additional geometrical structures on the mask for image distortion correction. Since this is generally a short range pattern correction (within $\sim 1 \mu\text{m}$ distance), significant geometry change [69] seldom occurs in this operation. Therefore, LVS verification performed before OPC can insure correctness of the post-OPC layout. However, mask boolean operations [34][35], which have been widely used for IC mask generation for special mask patterns, will dramatically change layout geometries and potentially even electrical connectivity. Moreover, in most common design flows, some mask layers, such as PW, Ldd, SiGe and CESL are not drawn GDS layers with geometries specified by designers during circuit layout. Rather, the patterns for these layers are computed by applying boolean mask operations (such as NOT, OR, SIZING) using the designer-drawn GDS layers as input, in order to reduce the complexity and data volume of designer-drawn layout. As a result, the original layout pattern is significantly modified by the mask boolean operations. That implies that LVS verification at the design stage might not adequately verify the on-mask patterns generated by these boolean operations.

With recent advances in deep submicron manufacturing technology, tech-

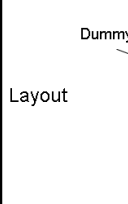
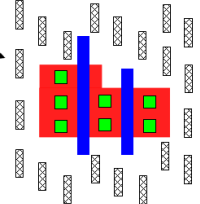
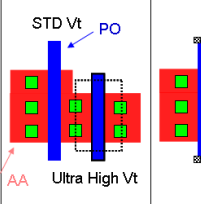
	Dummy fill operation	Mask Boolean Operation	OPC
Purpose	Pattern density balance	Reduce design effort, enlarge process marginality and transistor performance boost	improve the photo process marginality
Done	at design or foundry	at foundry	at foundry
Operation	Dummy cell filling in between active patterns	Pattern change using AND, NOT, OR and SIZING operation	Small patterns annotation on active pattern edge
Layout			

Figure 3.1: Comparison of layout pattern change due to various operations commonly applied to completed design GDS.

techniques such as strain engineering have been widely used to boost transistor performance [40]. In such techniques, additional mask layers such as SiGe and CESL layers are introduced, and these layers are generated by complex boolean operations to reduce design effort and potentially to safeguard the intellectual property related to these processes. Foundry mask booleans are made even more complex by the need to support customer-specified special-purpose devices which are not included in the main device offerings of the foundry's technology. Mask pattern issues rendering initial design LVS invalid are very difficult to detect by manual inspection, no matter whether these are caused by imperfect boolean design or simple human error in boolean specification. It is common practice to perform intensive visual inspection to verify the correctness of the mask generation algorithms, but this cannot provide adequate coverage, and frequently the effects of non-optimal Booleans are only detectable after the product has actually been fabricated in silicon. Moreover, for cases where a customer needs to alter the device characteristics (e.g. V_t position), the customer will collaborate with the foundry to modify the mask algorithm

to enable the new device formation, resulting in modification of the generation of each and every affected mask layer. Some of these changes increase the complexity of the logic operation equation (in some scenarios the total number of characters in a single mask algorithm exceeds 32K characters). To validate the correctness of the modified mask algorithm, various test patterns are created to confirm that the affected mask layers correctly implement the new customized devices.

In this paper, we propose Mask-versus-Schematic (MVS) verification, a novel design verification flow introducing an additional verification step after boolean operations have generated the final mask GDS. Instead of performing LVS on the original design layout as performed in the conventional design flow, we perform a MVS on the mask data which directly determine the patterns printed on the wafer. In other words, the layout produced by the mask algorithm is compared against an equivalent schematic netlist using a modified version of the Layout-versus-Schematic (LVS) runset. Since this Mask-versus-Schematic (MVS) verification is performed before the production of the actual reticles, any mismatches between layout and schematic generated by boolean operations can be corrected prior to the fabrication of costly reticles.

3.1 Background

3.1.1 Layout versus Schematic

A general custom-design verification flow is shown in Fig. 3.2. Once the physical layout for the particular circuit block is completed, a layout circuit extractor is invoked to determine the connectivity of primitives (MOS, diodes, resistors, capacitors) in the layout. Traditional LVS consists of two steps. First, a graph isomorphism program is used to compare the connectivity of the extracted netlist with the initial as-designed schematic. This program verifies that the two netlists have

identical connectivity by assigning primitives to the nodes of a graph and checking the connections to determine the extent of the connectivity match between the two netlists. Next, once connectivity equivalence has been verified, attribute matching is checked for each primitive attribute (e.g., capacitor, or resistor value, transistor W/L) [41]. Once the original design GDS file has passed this LVS verification, it is passed to the next step for dummy fill operations, which is shown in the traditional verification flow. The data sets to generate the reticles are generated after the mask boolean and OPC operations have been applied.

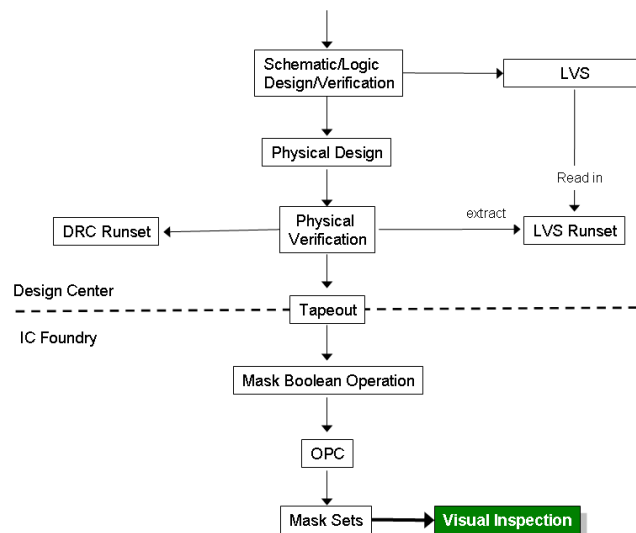


Figure 3.2: Conventional Flow from Design to Mask Making.

3.1.2 Mask Boolean Operation

Boolean operations are widely used for IC mask generation. When the GDS file of a design is sent to an IC foundry or mask vendor, it is necessary to first compute all the mask layers from the initial as-drawn GDS layers (referred to hereafter as the CAD layers). Note that the number of CAD layers contained in a GDS file is around 20 while a CMOS mask set typically includes at least 40 mask layers

(depending on the specific process technology). Some mask layers, such as PW, Ldd, and SiGe, are not included in the CAD layers, rather they are computed by applying boolean mask operations (such as NOT, AND, OR and SIZING). Fig. 3.3 illustrates the formation of a set of masks by boolean operations on a NAND gate. The PW and NLDD layers, not present in the CAD layout, are generated by boolean operation using the drawn layers NW, N+, and PO. In general, when a design is delivered from a design house to a IC foundry for production, the LVS and dummy fill operations are performed by the design house, and the computation of the final mask layers is performed by the foundry. Designers usually verify the final mask data by visual inspection through an internet-based verification tool offered by the IC foundry as shown in Fig. 3.2. Frequently, errors caused by mask boolean operations are not easily detected by such a simple human visual inspection. As an example, we compare the shapes produced by correct and erroneous boolean equations for the generation of the NLDD layer, as shown in Fig. 3.4, where the erroneous boolean is caused by a simple typographical error. The erroneous equation is not detected by traditional LVS, but it does not correctly define the silicon area to receive NLDD implantation, which will almost certainly result in a non-functional device. In addition, the boolean operations strongly depend on the process technology used, and differ among various IC design and foundry companies. Moreover, in advanced process technologies, the boolean equations required to generate the mask layers become more and more complicated, and verification is not easily performed in the conventional design flow.

3.1.3 Customized device using mask boolean operation

In general, IC foundries offer several general-purpose devices (such as standard Vt, high Vt, and low Vt, etc.) to fabless companies to enable a wide range of

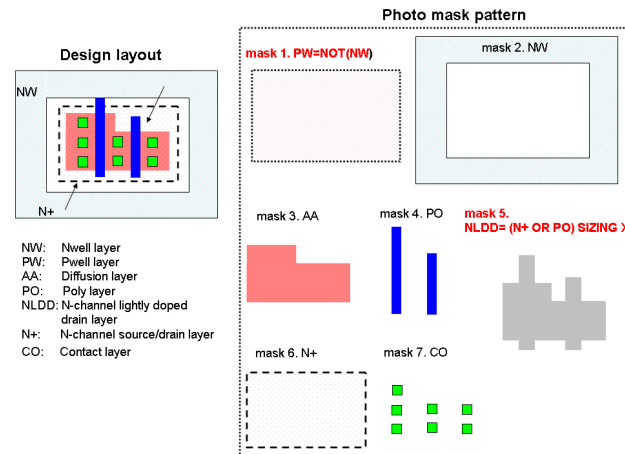


Figure 3.3: Mask layer formation by boolean operation. In general, P Well and NLDD layers are generated by mask boolean operations. Other layers are directly copied with only sizing operations.

	Correct	Typo
Mask Boolean Operation equation	5. NLDD= (N+ OR PO) SIZING X	5'. NLDD= (N+ AND PO) SIZING X
Generated Pattern		

Figure 3.4: Comparison of LDD mask layers generated by correct and erroneous mask boolean operation equations.

design choices [42][43]. For example, analog and digital circuits have somewhat different requirements. However, as customers increasingly demand more performance, lower power, increased functionality, and IP portability, the need for customized devices increases accordingly. However, IC foundries encounter serious challenges in providing customized devices to each fabless customer. Several fabless companies have devised customized mask boolean operations to create their own special-purpose devices such as ultra low, high and medium V_t to improve their product specification flexibility and meet rapidly changing customer requirements. Basically, the drawn layouts of such customized devices are similar to the layouts of

the standard transistor offerings of a given technology, but the customized devices use different combinations of the standard implant layers. Mask boolean operations which are a function of both specified layers and layer datatype are commonly used for generating such special devices. Referring to Fig. 3.5, the ultra high V_t nMOS, which might not be a standard device offered by a particular IC foundry, can be formed using an additional GDS layer and a boolean operation defined by the designer. In addition, as process technology continues to advance, the number and complexity of special devices correspondingly increase to address application-specific requirements. As a result, the entire algorithm for mask boolean operations required for special device definition may become substantially more complicated than the default mask-generation boolean operations offered by an IC foundry. Note that the special devices are formed at the mask boolean operation step of the design verification flow as shown in Fig. 3.2. Therefore, traditional LVS does not adequately verify such special devices.

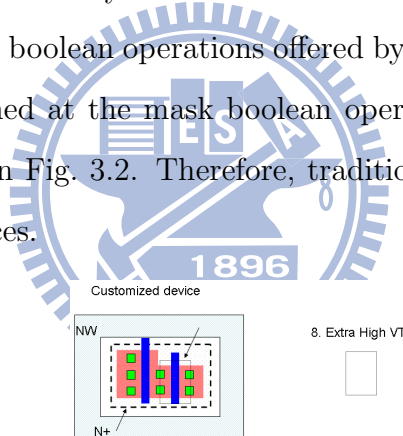


Figure 3.5: Customized device formation by boolean operation. The ultra high V_t device is generated by introducing a new layer and mask boolean operation.

3.2 Mask Versus Schematic Methodology

3.2.1 Overall Flow

To reduce exposure to potential errors during visual inspection of the mask algorithm output, and to automate the verification of increasingly complex mask generation algorithms, the MVS methodology utilizes the existing LVS runset that is traditionally used to perform physical verification during tapeout of the CAD GDS

to IC foundries. The main reason for using LVS to detect mask algorithm errors is that LVS is able to perform device and connectivity extraction based on the layer construction for comparison against the original design netlist. LVS performed on mask layout GDS provides the same degree of verification on the GDS generated by mask booleans. As shown in Fig. 3.6, two new procedures are added to the conventional mask fabrication flow.

First, the mask algorithm is converted to a physical verification runset through a script. The runset reads the tapeout GDS file and generates a virtual mask set GDS. GDS is a hierarchical format, whereas the format most commonly used to transfer mask data to mask shops is the non-hierarchical CFLT format [44]. The creation of a set of mask layers in hierarchical GDS enables verification by common commercial EDA tools, whereas such tools could not be used on the enormous CFLT files used to create the photomasks. Each virtual mask layer is generated according to the mask algorithm operation and is mapped to a unique pre-defined GDS layer which is not one of the CAD GDS layers shown in step 1. All the virtual mask layers generated from the runset are stored in a single GDS file. Next, the LVS runset is modified according to the changes in GDS layers and mask generation algorithms to create the MVS runset as shown in step 2. The MVS runset reads the new virtual mask set GDS and extracts a netlist for comparison with the initial design netlist, just as traditional LVS generates such a netlist from the CAD GDS. If the virtual mask set GDS is created correctly by the mask algorithm, the MVS run should produce an error free MVS output. If any errors are present in the mask algorithm, these errors will of course be present in the virtual mask set GDS. The MVS run will detect all violations of expected device construction requirements, missing or extra mask layers, as well as discrepancies from the design netlist such as missing connections or short connections caused by errors in interconnect layer generation.

As previously noted, the operations of dummy fill generation and OPC will change the geometry of layout patterns. In fact, MVS will detect errors resulting from pattern change during dummy fill because the dummy fill operation is performed prior to the mask booleans. OPC, as described in section ??, is a relatively short range pattern modification (within $\sim 1 \mu\text{m}$ distance), and seldom causes significant pattern geometry change. If desired, the impact of OPC can also be checked by inserting an MVS operation at step 3. However, the MVS execution time at step 3 will significantly increase due to the much more complicated patterns created by OPC.

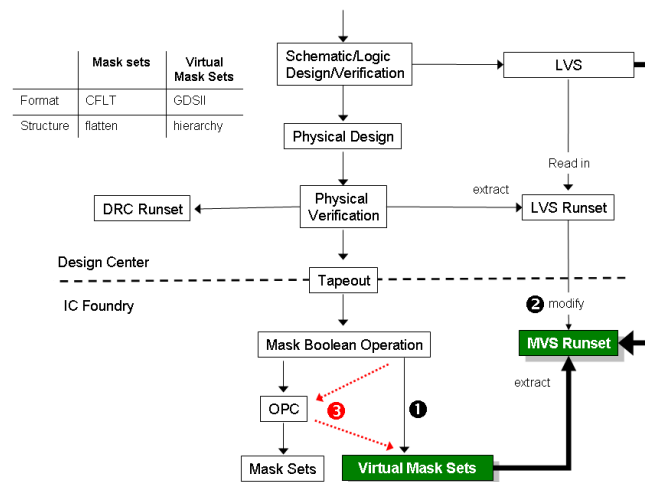


Figure 3.6: MVS Flow for Mask Generation Algorithm Verification.

To perform MVS verification, two programs are required. First, virtual mask set generation creates copies of the final mask layers in GDS format, and second, the MVS runset is altered from the original tapeout LVS runset used for physical verification of the CAD database.

3.2.2 Virtual Mask Sets Generation

The virtual mask set generation flow is shown in Fig. 3.7. There are several steps required to generate the virtual mask set. The associated algorithms are illustrated below:

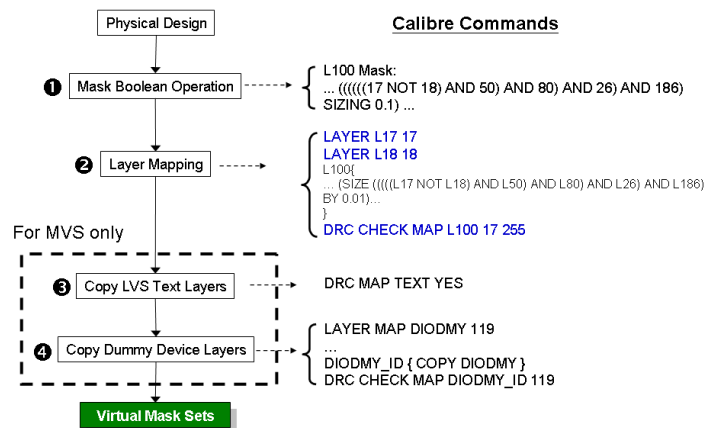


Figure 3.7: Virtual mask set generation flow.

3.2.2.1 Mask Boolean Operation

Virtual mask set generation is accomplished by a Calibre runset that performs the mask generation algorithm operations. An example mask generation algorithm is shown in step 1 of Fig. 3.7. L100 represents the mask layer and each number in the mask algorithm represents a GDS layer.

3.2.2.2 Layer mapping

A script is used to prepare the mask generation algorithm in Calibre Standard Verification Rule Format (SVRF), which converts the CAD GDS layers to virtual mask set GDS [46]. LAYER commands map the CAD GDS layers into internal Calibre layers. The layers are used inside Calibre command lines that mimic the mask generation algorithm. Each mask algorithm operation also includes a DRC

check where the result of the operation is redirected to the output GDS layer by the CHECK MAP command as illustrated in step 2. Each virtual mask layer is assigned to a unique output GDS layer and datatype. All the mask generation algorithms are included so that all virtual mask layers are combined into a single GDS output file.

The mask generation algorithm uses only GDS layers required for mask making. Two additional groups of data which are not part of the required mask layers, must also be copied into the virtual mask set GDS for MVS verification.

3.2.2.3 LVS TEXT Layers

LVS TEXT layers are not required for mask making but they are used to assist the net recognition and matching during MVS verification, and to help designers perform debugging if errors occur. The command in step 3 copies the LVS TEXT into the virtual mask set GDS. Fig. 3.8 illustrates the design layout of a NAND gate. There are four TEXTs (InputA, InputB, VSS, and Output) which must be copied to the Virtual Mask Set for the purpose of net recognition.

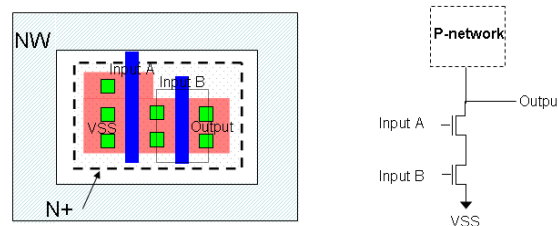


Figure 3.8: LVS TEXT Layers need to be copied to the virtual mask set GDS to facilitate net recognition and debugging.

3.2.2.4 Device Dummy ID

Device dummy ID layers are used for recognition of resistors, diodes and special transistor devices in LVS. To enable MVS to function, those device dummy

layers are copied from the tapeout GDS to the virtual mask set GDS during generation of the virtual mask set GDS as illustrated in step 4.

3.2.3 MVS Runset

The MVS runset is created by performing minor modifications to the Hercules LVS runset used for physical verification of the original CAD GDS file [47]. Four basic modifications are performed to enable the MVS runset to address the changes in layout due to mask algorithm operations as shown in Fig. 3.9.

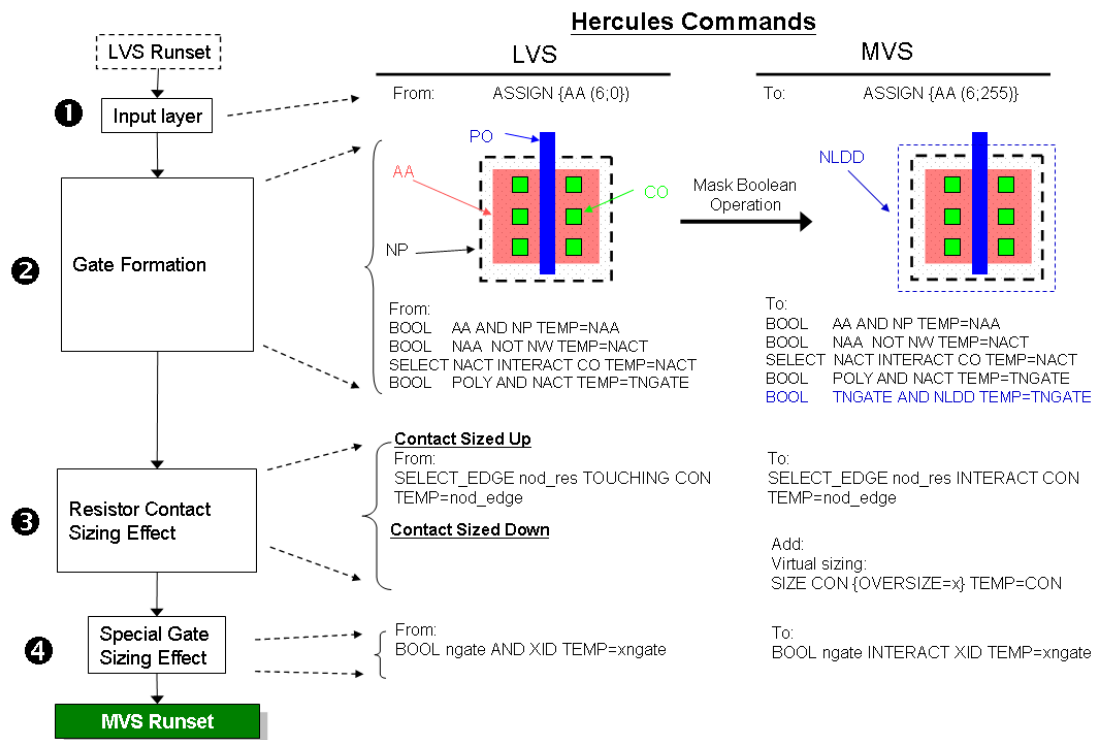


Figure 3.9: MVS runset generation flow using a simplified transistor formation example. The only difference between the GDS used for MVS and LVS is the presence of the NLDD layer in the MVS database. Real cases are much more complicated, as they contain many more additional layers generated by the mask generation algorithm.

3.2.3.1 *Input Layers*

The input layers are changed from the layer IDs and datatypes of the CAD GDS to those of the virtual maskset database as shown in step 1 of Fig. 3.9 following the output layer convention defined inside the Calibre runsets, e.g. Layer 6 datatype 0 is the original drawn GDS layer in the CAD database, and layer 6 datatype 255 is the corresponding virtual mask set layer.

3.2.3.2 *Device formation*

The method for definition of devices such as MOS, diodes, resistors, capacitors, etc for LVS extraction, needs to be slightly modified due to additional layers introduced during the mask boolean operations. In the nMOS example shown in Fig. 9 step 2, the nMOS definition for LVS is formed from the PO, AA, NP and CO layers with Hercules commands. However, for the MVS runset, an additional NLDD layer must be included in the nMOS definition.

3.2.3.3 *Contact Sizing Effect*

Some resistor formations in LVS expect contact layers on the resistor terminals to touch the device dummy ID layer that defines the resistor body. With the sizing effects which are usually present inside the mask algorithm operations, the resistor extraction commands must be modified accordingly. The contact may be sized up or down depending on the mask generation algorithm. When the contact is sized up, it overlaps the dummy ID layer as shown in step 3. To enable LVS extraction in this layout, the LVS code that uses the "TOUCH" command is modified to use the "INTERACT" command. When the contact is sized down, a gap is created between the contact and the dummy ID. To eliminate the gap for device extraction, the contact in the MVS code is sized back up virtually by a value 'x' based on the

gap between contact and dummy ID. Fig. 3.10 illustrates the photo mask patterns with contact sizing effects. It also shows the layout netlists extracted by LVS on both the CAD GDS and the virtual mask layer GDS resulting from the boolean mask operations. Using the original LVS runset on the virtual mask layer GDS is not effective because the device in the virtual mask layer GDS is not recognizable by the deck, and thus produces an empty cell netlist. To identify the device in the mask layer database, it is necessary to apply the modifications described above to create the MVS runset.

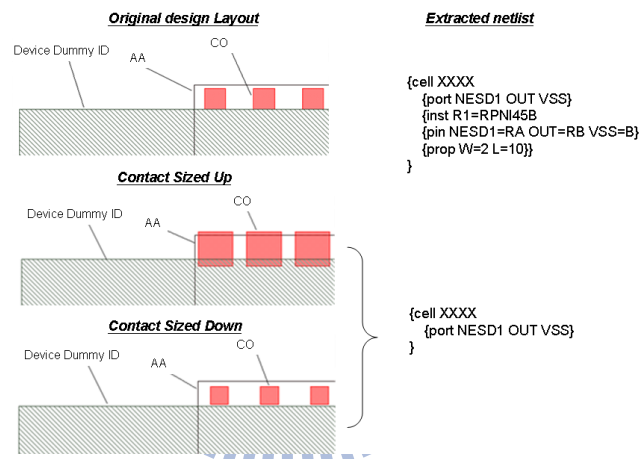


Figure 3.10: Resistor terminal contact placement in tapeout GDS.

3.2.3.4 Gate Sizing Effect

When gate sizing is employed in the mask algorithm, the gate device recognized by LVS through a special dummy ID requires modification for the MVS device extraction routine. In the CAD GDS, the gate dummy ID is typically drawn coincident with the shape drawn on the gate GDS layer. The use of gate sizing may cause misalignment between the gate shape and the gate dummy ID layer. Since the field poly is defined by the poly layer outside of AA, performing an AND operation to find the special gate device creates an open connection between field poly

and the gate because of a gap which appears between the gate and field poly as illustrated in Fig. 3.11. Without sizing the Dummy Gate ID, the single transistor will be incorrectly recognized as 3 different devices since part of the gate shape is not covered by the ID layer. To address this, the AND operation is replaced by the INTERACT operation.

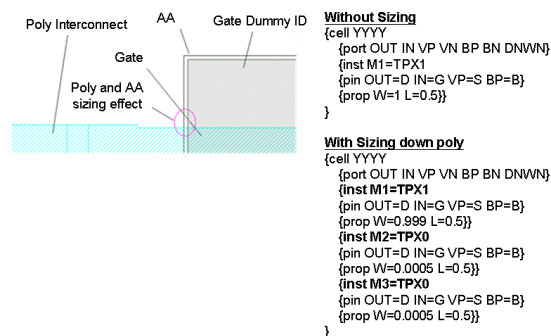


Figure 3.11: Sizing operations performed on the Poly and AA layers cause the gate dummy ID layer to misalign with the biased gate.

With the virtual mask set GDS generated by the Calibre runset and the MVS runset created from the Hercules LVS runset, the procedure for running MVS is the same as that for running traditional LVS. Illustrated in Fig. 3.6, the input files for a MVS run are: 1. Virtual Mask Set GDS 2. Netlist used for LVS verification. The results of a MVS run are reported in the same format as traditional LVS results, and hence errors can be investigated by similar debugging methods. Examples of MVS errors are shown in section 3.3 below.

3.3 Experimental Results

A test case to validate the MVS algorithm is described below. To simulate the effect of mask algorithm errors, three different bugs which are detectable through MVS verification were deliberately included in the mask generation algorithm.

3.3.1 Boolean Operation Error

This experiment demonstrates the ability to detect errors in mask algorithm boolean logic operations. Fig. 3.12 shows the mask algorithm with the correct logic operation, and the resultant virtual mask implant layer generated on the particular transistors in this layout.

$$\dots ((25 \text{ AND } 3) \text{ OR } \dots) \quad (3.1)$$

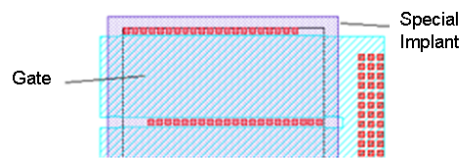


Figure 3.12: Correct mask algorithm for implant layer and the corresponding virtual mask set GDS with the generated implant layer.

An erroneous logic operation is purposely created in the mask algorithm where the AND operation is replaced by the NOT operation. This error causes the implant mask to not appear on the transistor as shown in Fig. 3.13.

$$\dots ((25 \text{ NOT } 3) \text{ OR } \dots) \quad (3.2)$$

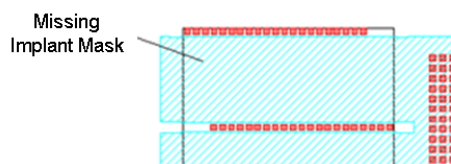


Figure 3.13: Incorrect boolean operation in mask algorithm and the resulting missing implant layer.

Fig. 3.14 shows the result of a MVS run based on the virtual mask set GDS generated from the incorrect mask algorithm equation 3.2. The missing implant on the transistor changed the device type and caused a mismatch in device types. The LVS cross-check feature can directly pin-point the devices in error.

Matched	Schematic Unmatched	Layout unmatched	Instance types [schematic, layout]
2	0	0	[R□, R□]
7	0	0	[S□, S□]
1	0	0	[T□, T□]
5	0	0	[T□, T□]
12	2	0	*[T□, T□]
4	6	0	*[T□, T□]
0	3	0	*[T□, T□]
0	4	0	*[T□, T□]
0	6	2	*[T□, T□]
31	21	2	*Total instances
17	14	3	*Total nets

Figure 3.14: MVS error from incorrect Boolean operation. Column 1 lists devices correctly matched between layout and schematic. Column 2 lists schematic devices not matched to any layout device. Column 3 lists layout devices which were not matched to any schematic device, and column 4 lists the respective device names of all matched and unmatched devices. The last 2 rows show the total number of devices and the total number of matched and unmatched nets respectively.

3.3.2 Braces Placement Error

The second experiment demonstrates the detection of an error in the placement of braces in the mask generation algorithm. The correct mask algorithm [equation (3)] does not generate a implant mask on the transistor shown in layout of Fig. 3.15.

$$\dots ((18 \text{ NOT } 3) \text{ NOT } 11) \dots \quad (3.3)$$

For this experiment, the braces inside the mask algorithm are relocated as shown in equation (4). This results in an additional implant mask layer being incorrectly generated on the transistors where no such layer should be present.



Figure 3.15: Correct mask algorithm for implant layer and the corresponding virtual mask set GDS with no extra generated implant layer.

$$\dots (18 \text{ NOT } (3 \text{ NOT } 11)) \dots \quad (3.4)$$

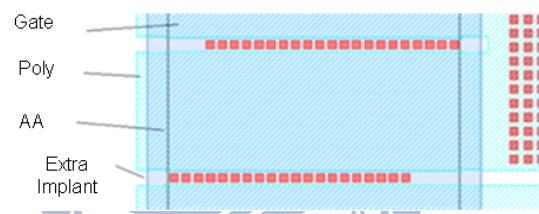


Figure 3.16: Incorrect braces placement in boolean operation and the resulting additional implant layer.

The result of the MVS run using the virtual mask set GDS in Fig. 3.16 and equation 3.4 is shown in Fig. 3.17. The additional implant on the transistor changed the device type and this error is detected as an un-matched device.

3.3.3 Sizing Error

A final experiment tests the ability to detect an error in a sizing operation inside the mask algorithm. Fig. 3.18 shows the original sizing value used to produce the correctly sized implant layer in the virtual mask-set layout generated by equation (5).

$$\dots ((18 \text{ OR } 90) \text{ SIZING } 0.1) \text{ SIZING } -0.2) \dots \quad (3.5)$$

Matched	Schematic Unmatched	Layout unmatched	Instance types [schematic, layout]
2	0	0	[R□, R□]
7	0	0	[S□, S□]
1	0	0	[TN□, TN□]
5	0	0	[TN□, TN□]
14	0	0	*[TN□, TN□]
3	0	0	*[TN□, TN□]
4	0	0	*[TP□, TP□]
6	0	0	*[TP□, TP□]
0	10	0	*[TP□, TP□]
42	10	0	*Total instances
19	12	6	*Total net

Figure 3.17: MVS error resulting from braces placement error.

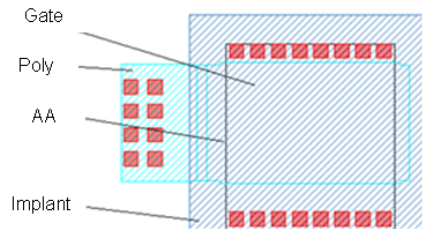


Figure 3.18: Equation shows the correct mask algorithm for implant mask generation and the virtual mask set GDS with the correctly sized implant layer.

The experiment uses an incorrect sizing value in the mask algorithm that produces a virtual implant mask layer smaller than the AA layer as illustrated in Fig. 3.19.

$$\dots ((18 \text{ OR } 90) \text{ SIZING } 0.1) \text{ SIZING } -0.6) \dots \quad (3.6)$$

The MVS result of the sizing error in equation 3.6 is shown in Fig. 3.20. The presence of the implant layer shape smaller than the corresponding AA creates two different transistor types at a single location. This results in a higher transistor count compared to the original schematic netlist.

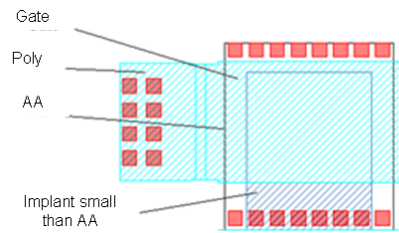


Figure 3.19: Incorrect sizing value in Boolean operation creates an implant layer shape layer smaller than corresponding transistor AA.

Matched	Schematic Unmatched	Layout unmatched	Instance types [schematic, layout]
2	0	0	[R□, R□]
7	0	0	[S□, S□]
1	0	0	[T□, T□]
5	0	0	[T□, T□]
14	0	0	*[T□, T□]
3	0	0	*[TP□, T□]
4	0	0	*[TP□, TP□]
0	0	14	*[T□, TP□]
8	2	14	*[T□, TP□]
0	0	12	*[TP□, TP□]
6	0	18	*[TP□, TP□]
50	2	58	*Total instances
17	14	60	*Total nets

Figure 3.20: MVS error from sizing value error.

Although the above experiment illustrates that MVS verification is able to detect sizing errors, detection of sizing errors is somewhat dependent on the severity of the mask error caused by the incorrect sizing value. If the sizing value is not significant enough to create a major difference in the mask produced, i.e. where the implant mask layer still encloses the AA in experiment 3.3.3, the MVS will not produce any verification error. However this type of dimensional error can be easily captured through a simple Post Logic(Boolean)-Operation Check(PLC) that relies on a DRC deck with known dimensions which are permitted to exist in the mask layout.

3.3.4 Qualification of MVS

A qualification procedure has been established to ensure that the methodology and all programs used for the MVS were fully functional prior to their use for validation of a real design. Test cases are drawn for all types of devices used in the process. Examples of all aforementioned possible errors described in 3.3 A to C were applied to ensure that they can be correctly identified from the MVS result. Once the MVS program has been validated against this suite of test cases, the program is qualified for use for real designs.

3.3.5 IP design experiment

A similar methodology was also tested on a real IP design with around 3.5 million transistors. The MVS verification computing platform consisted of a Dual Core AMD Opteron(tm) Processor 285 using Redhat OS with 1 CPU and 1 thread. Identical runtimes resulted from clean LVS and MVS runs despite the increased complexity of the MVS database. When MVS errors were present, the MVS runtime was only $\sim 1.3x$ that of the clean result. The following shows the results of the MVS experiment with 3 different types of Logic Operation errors.

	Matched devices	Unmatched Schematic instance	Unmatched Layout instance
Boolean Operation Error	1,407,280	2,134,743	366,745
Braces Placement Error	1,407,280	2,134,743	366,745
Sizing Error	3,285,961	256,062	684,612

Table 3.1: MVS statistics on a IP circuit with about 3.5 million transistors.

Chapter 4

A novel array-based test methodology for local process variation monitoring

As the feature size of devices scales down, the device variability imposed by each process step does not scale accordingly. As a result, the process variation of advanced process technology nodes has greatly increased and has become a critical factor in both IC design and manufacturing [67]. In order to design and manufacture in the presence of process variation, much research effort has focused on the areas of measurement, analysis, and modeling of variation during the past decade [68] [69] [70] [71] [72] [73] [74] [75] [78]. As the relative impact of process variation has continued to increase and become more randomized, research interest has shifted from *global process variation* toward *local process variation*, where the device characteristics within a close proximity vary randomly, since the sources of global and local process variation are different [67] [80]. This focus on local process variation has led to increased importance of array-based test structures, including decode logic, which are capable of characterizing a large number of individual transistors. However, the utility of such array-based test structures is limited by (a) the loss of accuracy due to voltage drop from parasitic resistance, and (b) the leakage current (including junction leakage) from the control circuitry. To overcome these challenges, we propose a novel test structure design and test methodology including *hardware IR compensation* to address the IR drop from parasitic resistance, and the

combination of *voltage bias elevation* and *leakage current cancellation* to eliminate both leakage currents from control circuitry and diode leakage related to hardware IR compensation.

In the typical measurement of global variation, a conventional test structure, a *PCM (process control monitor) testline*, is used on a wafer's scribe line. The PCM testline places its DUTs and IO pads in a straight line and uses four IO pads to measure each DUT. Thus, the height of a PCM testline is limited by the pad size, and so is the required spacing in a scribe line. To accurately measure local variation, a large number of DUTs need to be placed in close proximity and measured individually, or the measured results may misrepresent the process variation as a measurement outlier. However, using test structures such as PCM testlines to place a large number of DUTs within a small neighborhood is not feasible since the density of IO pads of such a structure would become too high (4 IO pads for each DUT) and would easily exceed the practical limit of a probe card.

In order to place a large number of DUTs close enough to one another and measure them individually, several array-based test structures have been proposed to share IO pads among DUTs and hence reduce the number of the required IO pads in between the DUTs [81] [82] [79]. These array-based test structures use row and column decoders to select an individual DUT in the DUT array and employ various techniques to address the IR drop imposed by the transmission gates on a DUT's selection path. The test methodology of [81] addressed the IR drop with some success, by employing an operational amplifier placed directly on the probe card, and utilizing an Agilent 4156 SMU in a force/sense circuit, similar to the hardware IR compensation technique we propose in this work. Though demonstrating the validity of this approach, this valuable work did not address the impact of the op-amp internal resistance on the voltage measurement error or the potential diode

leakage through transmission gates. Here-in, we explicitly consider the impact of the SMU internal resistance and introduce voltage bias elevation and leakage-current cancellation to minimize leakage through the transmission gates. The array-based structure of [82] suffers from large background leakage from the DUTs. [80] uses a ROM-like DUT-array design, which shares a common poly gate and a common drain among DUTs to avoid the usage of decoders. However, such a gate-sharing array design results in a DUT layout different from the device layout used in real products. Therefore, the measured process variation may not be representative of an actual product, which greatly limits the application of such ROM-like test structures. In addition, the ROM-like test structure may result in a large junction leakage current due to common drain/gate buses.

In this paper, we propose a novel array-based test structure utilizing decoders to access the DUT array, where all the peripheral circuits, such as decoders and latches, are implemented by I/O devices (thick gate oxide and long channel devices) and thus are not sensitive to process variation. In the proposed test structure, we develop a hardware IR-compensation technique to eliminate the impact of the IR drop imposed by the selection circuits. Also, we apply a voltage-bias-elevation technique to eliminate a possible negative voltage resulting from the IR-compensation hardware. Further, we develop a leakage-current-cancellation technique to reduce the background leakage when measuring a DUT's off-state current. Experimental results based on advanced process technologies demonstrate that the proposed array-based test structure can effectively measure hundreds of DUTs within a close proximity and the measurement accuracy of each DUT is almost the same as that measured by the traditional testline, on which only 8-20 DUTs can be measured. These experimental results also demonstrate the significant improvement of the measurement accuracy achieved by applying the proposed techniques. Also, compared

to a ROM-like array-based test structure, the measured results of the proposed test structure can indeed reflect the reality of a manufacturing environment. The DUTs used in the experiments include single devices for discrete transistor characterization, and paired, identical adjacent devices for measurement of local mismatch.

4.1 Background

4.1.1 Traditional Testline(PCM)

Fig. 4.1 shows the overall architecture of a conventional PCM testline, which consists of DUTs and IO pads arranged in a straight line. Each DUT is connected to 4 IO pads (one each for source, drain, gate, and bulk connections). Each IO pad is connected to the forcing and sensing node of a SMU (source/measure unit) during testline measurement through the probe card (Fig. 5.1). In this architecture, the voltage at the connection of forcing and sensing paths, commonly referred to as the *compensation point*, must be equal to V_{set} since no current runs through the sense path. Therefore, the IR drop caused by parasitic resistance from the tester to the compensation point is completely eliminated. However, the voltage at the drain node of the DUT may drop significantly because no voltage compensation exists between the compensation point and the drain node of the DUT. This results in a degradation of the measured current, especially when the width of the DUT's MOSFET is large ($W > 2\mu\text{m}$). This degradation of the DUT's measured current strongly depends on the parasitic resistance from the compensation point to the DUT. Four possible sources of parasitic resistances in a conventional PCM testline and their approximate values for the process technologies and test equipment of interest are listed below:

- Cable resistance from tester switch matrix to probe card < 1 ohm.

- Contact resistance between probe card needle and testline pad=1-20 ohm, depending on factors such as: (a) probe-card cleanliness and quality, (b) whether the top surface of the probe pads on the wafer is Al or Cu, and (c) queue time and storage ambient before probing.
- Metal routing resistance from testline pad to DUT source/drain. For a representative example, we consider 50 μm pad spacing, metal sheet resistance $R_s=0.2$ Ohm/square (a typical value for M1), and routing layout consisting of 3 μm wide metal for most of the routing distance in series with 3 parallel metal lines 0.1 μm wide and 3 μm long to connect to the DUT. The routing resistance will then be of order $0.2\text{Ohm} \cdot (25\mu\text{m}/3\mu\text{m} + 3\mu\text{m}/(0.1\mu\text{m} \cdot 3)) \sim 5\text{-}10$ ohm. Of course the exact value for a given PCM depends on the specific metal process and routing layout.
- Resistance from source/drain contact to active silicon. For example, for a contact process having contact resistance of 60 Ohm/contact and 7 parallel contacts to a DUT with 1 μm channel width, this resistance will be of order $60/7=8\text{-}10$ ohm, where the exact value depends on details of the contact and salicidation processes.

Typically, the total parasitic resistance from the compensation point to a DUT is approximately 1-30 ohms in a conventional PCM testline configuration, and in the worse case a significant error in the measured current can be generated. In addition, a large number of DUTs are required to monitor local process variation. However, this testline architecture can only contain around 8-20 DUTs in a $\sim 60 \times 2200 \mu\text{m}^2$ scribe line, which greatly limits the sample size of the measured DUTs and is hence not suitable for device variation modeling or process monitoring of advanced process technology nodes.

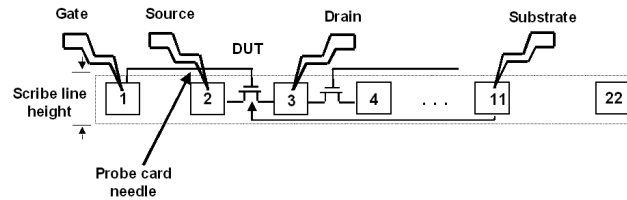


Figure 4.1: The configuration of a conventional PCM testline.

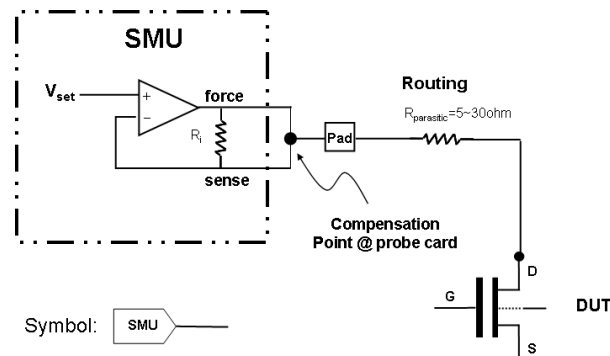


Figure 4.2: Voltage compensation mechanism used in a conventional PCM testline.

4.1.2 Transistor Array Test Structures and Adaptive Voltage Compensation

Test structures based on transistor arrays consistently must address two challenges: the IR drop along the electrical path to the DUT, and leakage current due to whatever DUT selection circuitry is employed. Here we briefly review key representative works in the field [11] [13] [14] [15] and describe past efforts to address these challenges in test structures utilizing transistor arrays. The test structure of [11] is array based, utilizing transmission gates, however its use of common gate and drain connections renders it susceptible to large leakage currents similar to the ROM-like architecture. The work of [81] addressed the IR drop by placing an operational amplifier directly on the probe card, and utilizing an Agilent 4156 SMUs in a force/sense circuit, similar to the hardware IR compensation technique we describe below, however [81] did not address the impact of the op-amp internal resistance

on the voltage measurement error or the diode leakage through transmission gates. Approaches to these issues will be discussed in our work in the sections that follow. The array-based structure of [82] suffers from large background leakage from the DUTs themselves. The voltage bias elevation and leakage current cancellation techniques we introduce substantially alleviate the leakage current issues suffered by the works list above. An innovative approach to voltage compensation was introduced by [83], where-in iterative adaptive voltage compensation was employed to compensate for the IR drop to the DUT. We describe this approach further below.

For array-based test structures, the IR drop from the compensation point to the DUT may become even larger due to the extra parasitic resistance of the added transmission gates and routing paths used for the decoding and selection scheme. To reduce the measurement error caused by this large IR drop, an adaptive voltage-compensation scheme was proposed in [83], which utilizes two SMUs for each node of the measured MOSFET as shown in Fig. 4.3. For each terminal of the DUT, one SMU (e.g. SMU1 for D, SMU3 for G, SMU7 for S, and SMU5 for sub) is used for forcing the voltage, which is then sensed by the other SMU (e.g. SMU2 for D, SMU4 for G, SMU8 for S, and SMU6 for sub). Fig. 4.4 lists the adjustment algorithm for finding a proper compensation voltage at each measured node. The approach of this algorithm is to incrementally increase the forcing voltage each time by a small step, and stop when the sensed voltage is equal to the reference voltage. The number of sweep steps of the forcing voltage determines the runtime and the accuracy of the adaptive voltage-compensation scheme. A small sweep step may result in a more accurate measurement, but it will also require longer runtime.

This adaptive voltage-compensation scheme is straightforward and easy to implement. However, it requires 8 SMUs per DUT and results in a high test-hardware overhead. Also, its adjustment algorithm may require too much time to

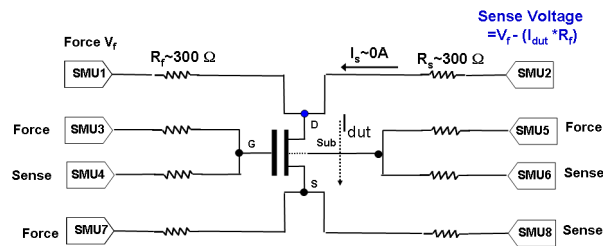


Figure 4.3: The schematic of the adaptive voltage compensation for one DUT.

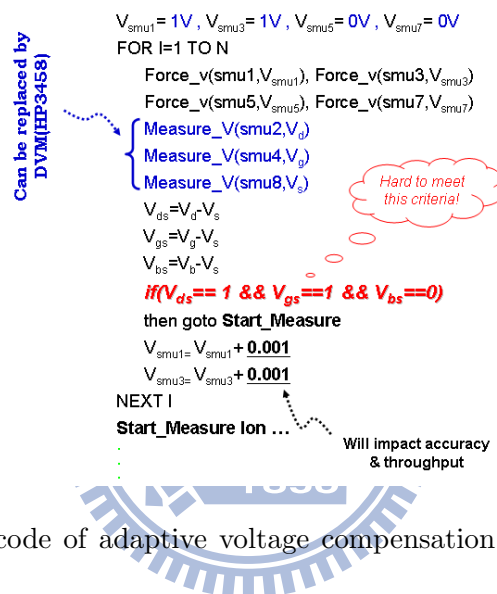


Figure 4.4: Pseudo code of adaptive voltage compensation algorithm for Ion measurement.

search for the proper compensation voltages and may even be unable to converge. In fact, this adaptive voltage-compensation scheme only converges easily when the number of measured nodes is equal to two, such as for a diode or resistor. The difficulty of convergence increases when the number of compensation nodes increases. However, most FET characterization such as SPICE modeling is performed on discrete transistors, meaning that all four FET terminals (drain, gate, source and bulk) need to be compensated. Therefore, the adaptive voltage-compensation scheme may not be practical for the applications described in this paper.

4.1.3 ROM-like transistor array

Several test structures using ROM-like transistor arrays have been proposed in the past to measure a large number of DUTs within a close proximity [72] [80]. A ROM-like transistor array requires no periphery circuit for DUT selection and can avoid the extra parasitic resistance of the transmission gates as described in the previous subsection. Fig. 4.5 shows a ROM-like transistor array design in which transistors on a given column share a common drain bus, while each row of transistors shares a common gate bus. All transistors' source and bulk nodes are tied together with wide metal layers to minimize IR drop. The FET array layout permits each individual transistor to be accessed without periphery circuitry. For example, while measuring FET T11, column 1 is biased and other drain columns are floating to reduce the diode leakage of the other transistors of columns 2 through 10. The voltage on other gate rows connected to T11 are biased at 0 V or a small negative voltage to turn off transistors and minimize the leakage in column 1 [80]. However, this design and biasing scheme still exhibits a major DC leakage current problem due to the use of common gate and drain busses, which results in an incorrectly measured value of V_t which is lower than the correct value obtained from a conventional single transistor testline. Traditionally, ROM-like open/short test structure arrays have been commonly used as test vehicles for defect monitoring during yield ramp [72]. The only ROM-like array structures which avoid this issue are open/short test structures evaluated by a strict pass/fail criteria, where the criteria for a short is a current level of order at least $\sim \mu\text{A}$. In addition, the layout style of a ROM-like transistor array, as shown in Fig. 4.6, cannot represent the end shortening and rounding effects of a poly gate used in actual product devices. Therefore, its measurement results may substantially deviate from the reality encountered in a product circuit.

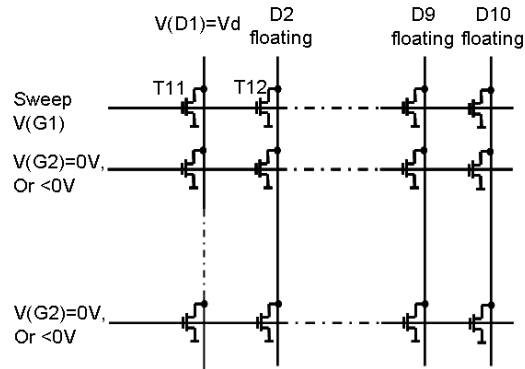


Figure 4.5: The architecture of a ROM-like transistor array using shared common gate and drain buses [80].

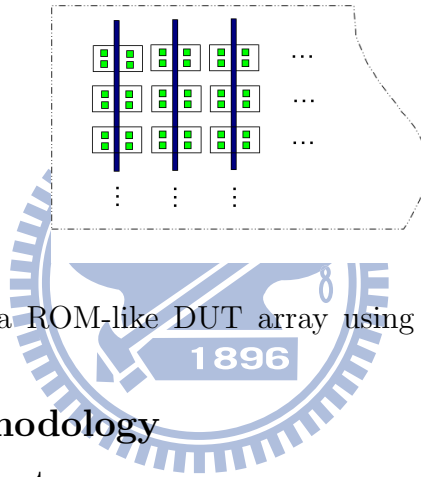


Figure 4.6: Layout of a ROM-like DUT array using straight poly lines for gate connections [80].

4.2 Design Methodology

4.2.1 Design Architecture

Fig. 4.7 shows the proposed transistor array with $4 \times 64 = 256$ test units. Each test unit consists of eight transmission gates and one DUT. The DUT (usually a MOSFET) can be measured by selecting the corresponding test unit through the X- and Y-decoders. In total, 8 address inputs (X1-X2 and Y1-Y6) are used to select the 256 test units. Each test unit is designed with a stand-alone connection to its DUT, which means each of the drain, gate, source, and bulk nodes is connected to a SMU. This stand-alone connection can be used for the measurement of I_{off} , I_{boff} , I_{goff} , or body bias of a DUT. Therefore, this test-unit design can also be modified to adapt to any type of device such as diode, resistor, pMOS, or nMOS,

as long as the number of terminals connected to pads is less than or equal to four. The terminals F1-F4 and S1-S4 are the SMU's forcing and sensing ports which are connected to the drain, gate, source and bulk of a DUT, respectively. The proposed test structure with 256 DUTs can fit into a regular 1x22 pad frame used for a standard PCM testline. Also, the array size of the proposed test structure can be extended to $16 \times 64 = 1024$ to increase the sample size for statistical SPICE modeling or other applications requiring large sample size. However, the extended array cannot be placed in the scribe line for production variation monitoring due to its larger area.

Fig. 4.8 shows the test-structure layout in a $60 \times 2200 \mu\text{m}^2$ region, where the size of each test unit is $30 \times 10 \mu\text{m}^2$. Note that the shaded square in Fig. 8c, representing the $\sim 10 \times 10 \mu\text{m}^2$ area immediately surrounding the DUT, may be designed with a layout representing a typical circuit environment, or with a layout deliberately varying one or more layout parameters. This can be accomplished without compromising the low series resistance of the wiring to the DUT terminals. Because most stress-related layout effects occur over length scales less than $\sim 5 \mu\text{m}$, the layout-induced stress on the DUT can easily be made identical to a typical circuit layout. In this proposed test structure, the circuit-under-pad (CUP) design technique is used to increase the number of DUTs placed in this $60 \times 2200 \mu\text{m}^2$ region. Note that this CUP design technique requires a process with at least four metal layers. By using the CUP design, the bottom few metal and via layers (metal-1 to metal-4) in the pad frame are removed. The active circuitry including test devices are all placed under the bond pads. Using the CUP design, the number of placed DUTs is increased by a factor of 2.5. In addition, all periphery circuits, such as latches and decoders, are designed with I/O devices rather than core logic transistors so that their background leakage can be reduced and their performance will not

be affected by the process variation of the advanced process under study. For the technologies under study, $V_{dd}=1.0V$ for the core logic transistors, and $V_{dd}=2.5V$ (or optionally $1.8V$) for the IO devices. In the examples discussed in this work, we focus on the case where the DUT is a core logic transistor, but the techniques are equally applicable to the case where the DUT is a $2.5V$ (or $1.8V$) IO device as long as the periphery circuits are overdriven to $3.3V$ (or $2.5V$).

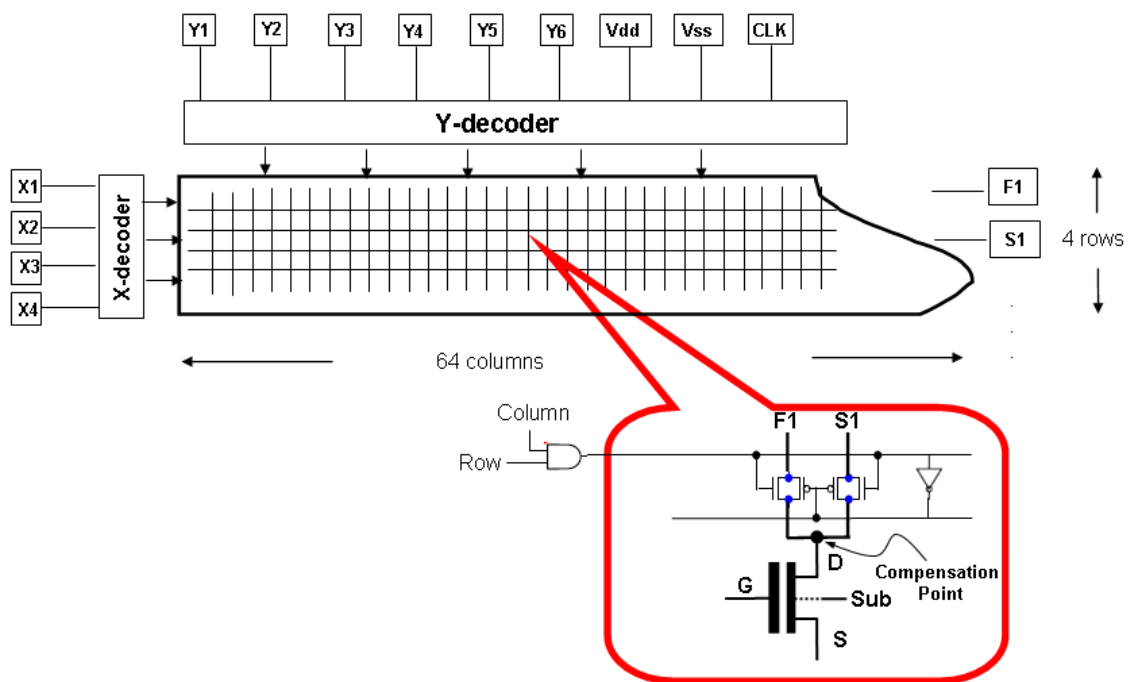


Figure 4.7: Architecture of the proposed transistor array with 4×64 DUTs for scribe-line-compatible footprints.

4.2.2 Hardware IR Compensation

A hardware IR-compensation technique is used in the proposed array-based test structure to reduce measurement error caused by the large parasitic resistance of the added transmission gates and routing paths. The key concept of this hardware IR compensation is to separate the forcing and sensing paths originally connected at the probe card (as shown Fig. 5.1), and re-connect them at a position very close

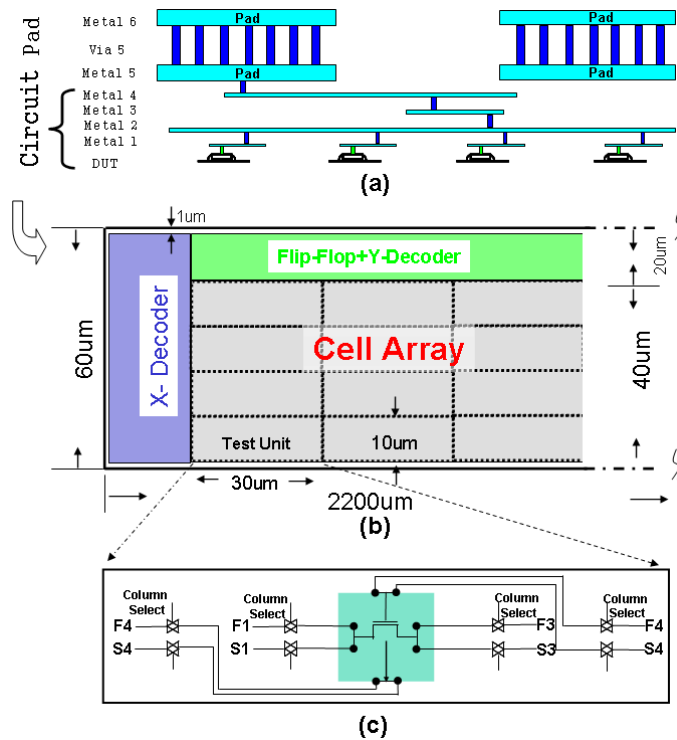


Figure 4.8: Layout of the proposed test structure using circuit-under-pad design (CUP) for scribe-line compatible footprints. (a) cross-section view of CUP design, (b) top view of the proposed test structure, and (c) schematic inside each test unit.

to the DUT. Such a connection can reduce the parasitic resistances between a compensation point and its corresponding DUT. In our test structure, the approximate distance between the compensation point and DUT is less than 2 μm. Thus, the parasitic resistance between compensation point and DUT is significantly lowered. For example, using typical metal resistance values, the parasitic resistance for each connection can easily be reduced to ~ 0.2 Ohm which is substantially lower than the 1 to 30 Ohm for a conventional PCM test-line as described in Fig. 2. Fig. 4.9 shows the whole configuration of the proposed setup, which contains the enhanced Kelvin connection, Agilent 407X SMUs, and a redesigned compensation point. All the DUT's terminals including drain, source, gate and bulk are compensated by this enhanced Kelvin connection to ensure measurement accuracy. The measurement

error for the configuration of Fig. 4.9 can be calculated by the following equations. We note that because R_i is much larger than R_s and R_f , to simplify the calculation, the negligible current flowing through R_i is taken to be zero.

$$V_{sense} = V_{set} \quad (4.1)$$

$$\begin{aligned} Error &= V_{set} - V_{dut} \\ &= V_{drops} \\ &= V_{dropf} \cdot \frac{R_s}{R_s + R_i} \\ &= I_{dut} \cdot R_f \cdot \frac{R_s}{R_s + R_i} \end{aligned} \quad (4.2)$$

As Equation 4.2 shows, the compensation error increases as I_{dut} increases. In order to address this compensation error when I_{dut} is high, we can either reduce the parasitic resistances R_s and R_f by layout engineering, or increase the internal resistance R_i at the SMU. To reduce R_s and R_f , we can increase the size of a transmission gate such that its channel resistance is reduced, or increase the metal routing width. However, a larger transmission gate results in larger leakage [82], and wider metal routing requires more layout space. Therefore, in our hardware compensation, we choose to increase the SMU's internal resistance R_i to limit the measurement error.

Increasing R_i lowers the difference between V_{dut} and V_{set} . Typically, the SMU's internal resistance in an Agilent 407X tester is a few k Ohm. For instance, with V_{dut} of 1V, the error percentage with a 10k-Ohm R_i and a 300-Ohm R_s is approximately $870\text{Ohm} \cdot (I_{dut}/1\text{V}) \%$. This error percentage can be reduced to $45\text{Ohm} \cdot (I_{dut}/1\text{V}) \%$ if R_i is increased to 200k Ohm. Since the maximum I_{dut}

for most typical device measurements is approximately 2mA, the worst case error percentage can be improved from 1.57% to less than 0.09%, if R_i is increased from 10k Ohm to approximately 200k Ohm. The error percentage slightly increases with I_{dut} for wide device width ($>2\mu\text{m}$). However, a 2mA maximum current level is adequate for most applications such as SPICE modeling, process diagnostics, stress, DFM and variation characterization. For a worst case of DUT measurement with $I_{dut}=10\text{mA}$, $R_i=200\text{K Ohm}$ and $R_s=R_f=300\text{ Ohm}$, the error will be smaller than 0.44%, which is superior to either a conventional PCM or adaptive voltage compensation. However, increasing R_i results in larger voltage convergence time. This technique should, therefore, be used with caution.

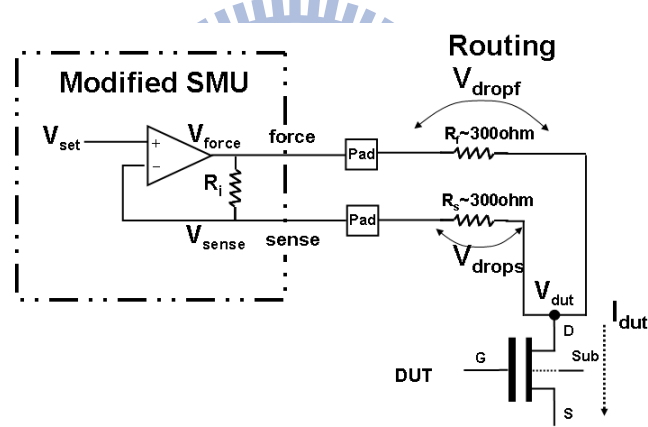


Figure 4.9: Schematic of the proposed hardware IR-compensation mechanism.

4.2.3 Voltage Bias Elevation for Measuring Ion

In the previous subsection, we introduced the use of the proposed hardware IR compensation to achieve high measurement accuracy, especially when I_{dut} is high. However, as shown in Fig. 4.10, the proposed IR compensation technique may create a negative node V_{sx} beside the transmission gate on the force path of the DUT's source side. During the IR compensation, the value of V_{sx} depends on I_{dut} and the parasitic resistance of metal routing and transmission gates, R_s . V_{sx} is in the range

of ~ -0.2 to ~ -1.0 V in the example of Fig. 4.10. Thus, if I_{dut} or R_s become too large, V_{sx} at the transmission gate's input node will become a large negative voltage, which may turn on the diode from source to substrate on the transmission gate and result in malfunction if V_{sx} is below -0.6 V.

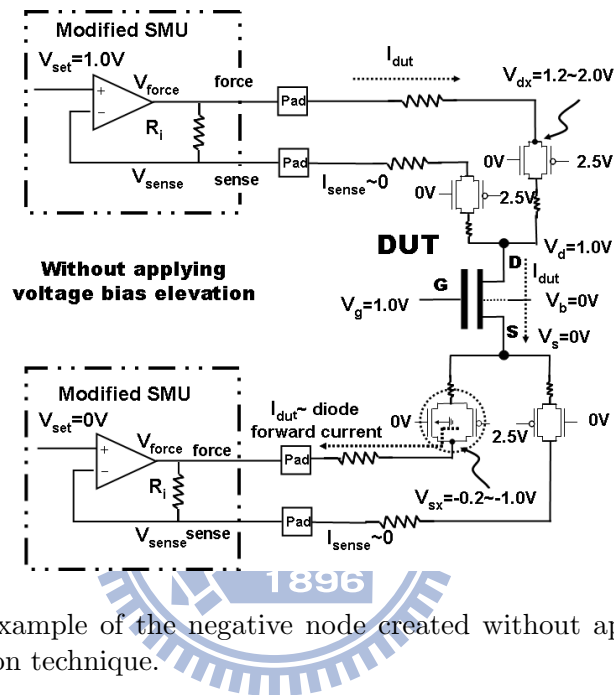


Figure 4.10: An example of the negative node created without applying the proposed voltage bias elevation technique.

Considering the case of I_{on} measurement for a core logic transistor with $V_{dd}=1.0$ V for example, before the OP Amp settles, the voltage at the drain compensation point would initially be below 1.0 V due to the IR drop. V_{force} would continue increasing until the voltage of the compensation point V_{dut} reaches 1.0 V. V_{force} would be elevated to approximately $1.0V + I_{dut} * R_f$. This does not present a problem on the forcing path of the DUT's drain side as long as V_{dx} is still below V_{dd} (i.e., 2.5 V). However, it would cause a negative node on the force path of the DUT's source side. The source-voltage setting of the DUT is 0 V for I_{on} measurement. The voltage at the input of the transmission gate would thus be decreased below the diode's turn-on voltage (about -0.6 V) due to the parasitic resistance of metal

routing and transmission gates, especially when I_{dut} is high. This would turn on the transmission gate's drain to bulk diode. As shown in the following equation, V_{sx} , and a corresponding "elevation" voltage V_{elv} , can be roughly calculated given I_{dut} , the diode's turn-on voltage V_{dt} , and the transmission gate resistance R_{TG} .

$$\begin{aligned} V_{sx} &= I_{dut} \cdot 2 \cdot R_{TG} > -V_{dt} \\ V_{elv} &= -V_{sx} \end{aligned} \quad (4.3)$$

If $V_{sx} < -V_{dt}$, the current measured by the SMU would be the diode's forward biased current, not the DUT current. To prevent this, the voltage bias of all the DUT's terminals are elevated to a positive voltage V_{elv} during I_{on} measurement to eliminate the negative voltage at the source path. This bias voltage V_{elv} is applied to V_{set} (thus also increasing V_{dx}), V_g , V_d , V_s , and V_b . The elevated voltage does not affect the electrical behavior of the measured DUT because the same voltage elevation is applied to all terminals concurrently. In addition, the power supply of the periphery circuitry can be overdriven to 3.3V to enlarge the compensation margin, i.e 3.3/2.5V VDD for a 2.5/1.8V I/O process. Figs. 4.10 and 4.11 respectively, show an example without and with application of voltage bias elevation.

4.2.4 Leakage-Current Cancellation for Measuring I_{off}

Another important MOSFET parameter to be measured is the off-state leakage current I_{off} . When measuring current, a SMU senses not only the DUT current, I_{dut} , but also the leakage current from periphery circuitry. For I_{on} measurement, the leakage current from peripheral circuitry does not affect the measurement accuracy since this leakage current is much smaller than I_{on} . However, this leakage

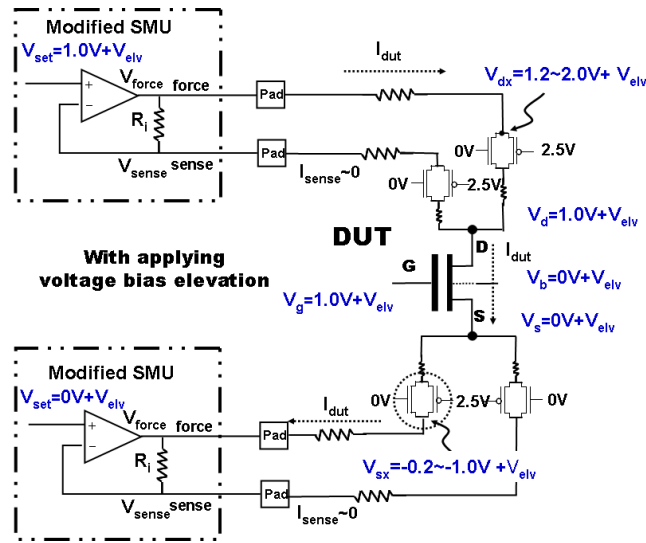


Figure 4.11: No forward biased current on the transmission gate is generated after applying voltage bias elevation ($V_{elv} = 0.5V$).

current can significantly affect the accuracy of I_{off} measurement. The sources of this leakage current include (1) the leakage from peripheral circuitry transistors and (2) the leakage from the transmission gates on selection paths. The second of these typically dominates the total leakage current. The leakage current from the transmission gates may result from the N+/PW and P+/NW junction leakage, and gate to drain leakage on both the pMOS and nMOS of the transmission gates. There are 64+4 leakage current paths created by the transmission gates in the proposed test structure as shown in Fig. 4.12 (64 gates for each column and 4 gates for each row).

In order to reduce the current from these leakage paths, we use I/O devices (having relatively thicker gate oxide and longer channel length) for designing the periphery circuitry. Also, we propose a leakage-current-cancellation technique for our array-based test structure. This leakage-current cancellation elevates all the DUT's terminals to an optimal voltage, similar to the voltage elevation technique described in the previous subsection. The operating principle of this leakage-current cancellation is to set the voltage of the DUT's drain node such that leakages from the

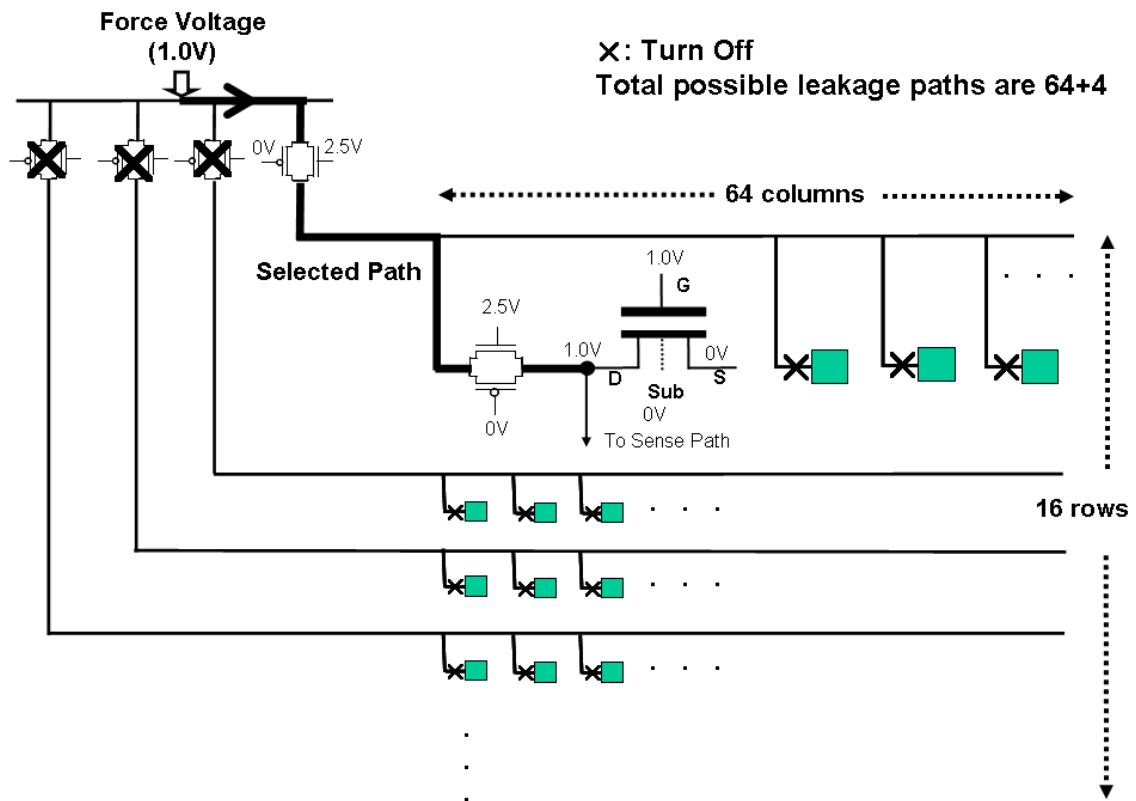


Figure 4.12: Possible leakage paths in the proposed array-based test structure. The thick line indicates the selected path. The background leakage current includes the leakage from both the selected and unselected paths.

nMOS and pMOS of each transmission gate on the 68 leakage paths can be balanced. Figs. 4.13 and Fig. 4.14 respectively illustrate the leakage current before and after employing the voltage elevation technique. Before voltage elevation, the voltage difference from the DUT's drain to both the nMOS gate and the pMOS substrate is 1.5V, but the voltage difference from the DUT's drain to both the nMOS substrate and the pMOS gate is only 1.0V. This unbalanced voltage difference may result in a large current on this transmission gate as shown in Fig. 4.13. After the voltage of the DUT's drain is elevated to an optimal value, the voltage differences from the DUT's drain to each nMOS or pMOS gate, or to the substrate, are all equal, such that the leakage currents from this transmission gate cancel out one another

as illustrated in Fig. 4.14.

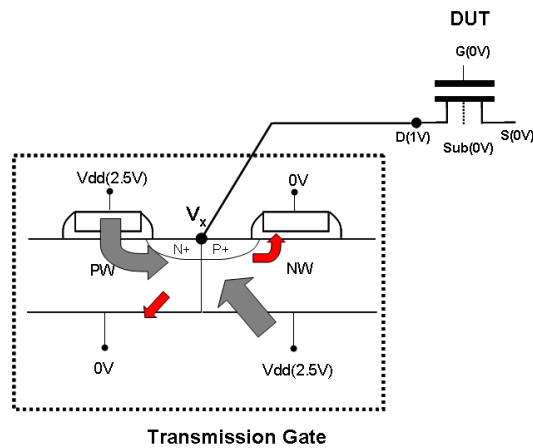


Figure 4.13: Larger background leakage due to unbalanced leakage paths before applying voltage bias elevation.

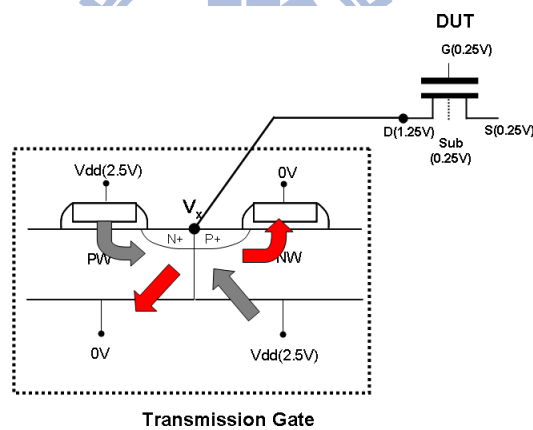


Figure 4.14: Reducing the leakage current by balancing the leakage paths with optimized voltage bias elevation.

If all transmission gates were perfectly fabricated, all nMOS and pMOS should be completely symmetric. In this ideal case, the optimized voltage at the DUT's drain should be half of the transmission gate V_{DD} , i.e., 1.25V in our example. However, in reality the fabrication of each transmission gate is also affected by process variation. Thus, when measuring I_{off} , the values of both V_{DD} and the elevated voltage at the DUT's terminals may be swept to find an optimal voltage

to minimize the leakage current from the 68 leakage paths during the measurement. The background leakage can be optimized by offsetting the drain voltage slightly from half V_{DD} . Fig. 4.15 plots I_{off} as a function of the drain voltage offset from half V_{DD} for the 64 DUTs in the array for one wafer (9 die tested on each wafer, with 1 array in each die). Typically, the minimum DUT off current can be obtained at a drain voltage fairly close to half V_{DD} . In this example, the sweep confirmed that the optimal voltage was quite close to half V_{DD} , as I_{off} is observed to significantly increase with a non-zero offset voltage. However, the optimal drain voltage might not be exactly half V_{DD} for all processes. It might be slightly above or below half V_{DD} for different processes depending on the difference between the NMOS and PMOS gate and junction leakage in the transmission gates, and sweeping the offset voltage as done in Fig. 4.15 permits confirmation or correction of the optimal offset voltage. Of course depending on the range and resolution of the offset voltage sweep, additional test time will be required to obtain this data. In this example, based on the data of Fig. 4.15, a voltage of half V_{DD} was determined to be adequate. We further emphasize that because long-channel thick-oxide IO transistors are used in the transmission gates, the offset voltage is relatively insensitive to process variation. In practical applications, a quick confirmation of the optimal offset voltage can be obtained with two additional I_{off} measurements, i.e. for positive and negative values of a single offset voltage. Further adjustment of the offset voltage need only be performed if it is shown to be necessary by these two measurements, in which case the offset voltage can be adjusted iteratively to minimize I_{off} to the desired precision, at the cost of 2 additional I_{off} measurements for each successive iteration. Because the need for offset voltage adjustment results from leakage imbalance in the periphery circuits, and these are long wide IO devices relatively less susceptible to local process variation, if such adjustment is necessary, it is likely to only be needed

once for each DUT array. Local process variation causes variation in I_{off} for different DUTs in the same array, but it is unlikely to affect the offset voltage which must be applied to the periphery to minimize the measured I_{off} value.

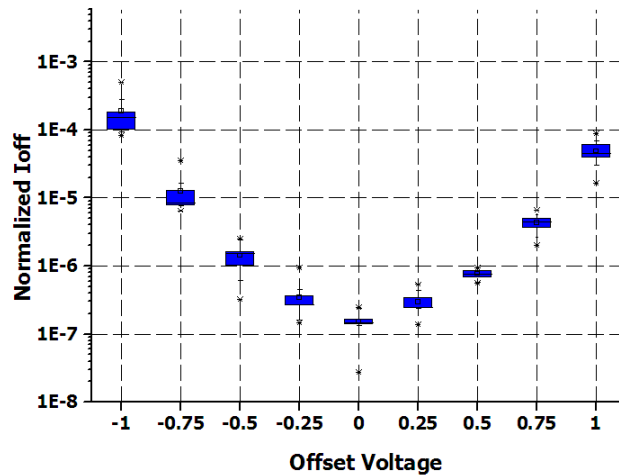


Figure 4.15: Background leakage reduction by offset voltage from half V_{DD} for the 64 DUTs in the array for 1 wafer (9 die/wafer with 1 array/die).

4.3 Experimental Results

4.3.1 Proposed Test Structure vs. Traditional PCM Testline

In the first experiment, a traditional PCM testline is fabricated adjacent to the proposed array-based test structure in a mature, relatively old process technology. In traditional PCM testlines, several types of devices with different dimensions are fabricated, such as MOSFET, RC, RS, and diode. We choose a DUT from the proposed DUT array closest to the DUT with the corresponding device type and dimension in the PCM testline, and then measure both of these two DUTs. The electrical properties of these two DUTs should be similar since the local process variation of this mature process technology is small.

Fig. 4.16 plots the drain current (I_d) versus the gate voltage (V_g) of the chosen

DUT measured by each test structure. Curve A represents the I_d measured by the traditional PCM testline, which is considered as the reference measurement result. Curve B represents the I_d measured by the proposed array-based test structure. As shown by Curves A and B, the I_d measured by the proposed test structure closely matches the I_d measured by the traditional testline when the normalized I_d is smaller than 10^{-3} (A.U), which covers the I_d range for measuring both the threshold voltage V_{th} and the saturation current I_{on} . Thus, the result shown in Fig. 4.16 demonstrates the measurement accuracy of the proposed array-based test structure for measuring V_{th} and I_{on} . Note that the I_d range for measuring I_{on} is in general below 1mA for advanced technologies. The proposed test structure can allow I_d measurement up to 2mA.

Next, we fabricate both test structures on an advanced, newly developed process technology and measure 9 traditional PCM testlines and 9 replicates of the proposed test structure over a wafer as shown in Fig. 4.17, which represents the typical sampling distribution of measured PCM testlines. On each chip, the proposed test structure is placed adjacent to the measured PCM testline. Fig. 4.18 plots I_{on} vs V_{th} for the DUTs measured from each PCM testline and from the proposed test structure. As shown by Fig. 4.18, the I_{on} and V_{th} measured by PCM testlines are all within the distribution of the I_{on} and V_{th} measured by the proposed test structure, meaning that the measurement accuracy of the proposed test structure is very close to that of the traditional PCM testlines. In addition, by using only the PCM testlines, the extent of the global variation on this wafer can be roughly observed, which is also one of the objectives of traditional PCM testlines.

However, the traditional PCM testlines fail to capture the local process variation. Fig. 4.19 plots I_{on} vs V_{th} of DUTs measured from only one PCM testline and its adjacent proposed test structure, demonstrating that the variation of the DUT's

I_{on} and V_{th} within the proposed DUT array is significant, but the PCM testline can only measure one sample from this distribution. This result further shows the necessity of an effective array-based test structure to measure a large number of DUTs in close proximity, such that local variation can be accurately characterized. In Fig. 4.20, we plot V_{th} vs DUT column (X axis) for all four rows of the structure. No spatial dependence is evident, indicating that the data truly characterizes the local random variation of the process.

4.3.2 Effectiveness of Hardware IR Compensation

In the following experiment, we measure the same DUT using the same proposed array-based test structure as that in Fig. 4.21 but without applying the hardware IR compensation introduced in Section 5.2.2, i.e. we use the SMUs' connection shown in Fig. 4.9, but the internal resistance of the SMUs remains unchanged. In Fig. 4.21, Curve C represents the I_d measured by the proposed test structure without hardware IR compensation. As Curve C shows, its measured I_d matches the reference result in the sub-threshold region. However, its measured I_d deviates from the reference result when the normalized I_d is larger than 10^{-4} , which falls in the current range required for measuring I_{on} . This measurement error results from the parasitic resistance of the transmission gates on the selection paths. Thus, without applying hardware IR compensation, the measured error of I_{on} can be quite large.

Fig. 4.22 further illustrates this error in I_{on} by plotting I_{on} vs. V_{th} for each DUT measured by the proposed test structures with and without applying hardware IR compensation. As shown in Fig. 4.22, the V_{th} distributions measured with and without hardware IR compensation are similar (see their X-coordinates). However, the I_{on} values measured without hardware IR compensation are significantly lower than those obtained using hardware IR compensation (see their Y-coordinates).

This is because the I_d for measuring I_{on} is much higher than that for measuring V_{th} and may result in a significant IR drop if no IR compensation is applied. The result shown in Fig. 4.22 again demonstrates the importance of applying the proposed IR-compensation scheme.

4.3.3 Effectiveness of Leakage-Current Cancellation

As discussed in Section 4.2.4, the proposed leakage-current cancellation can significantly reduce the background leakage from transmission gates by elevating the voltage at each terminal of a DUT to an optimized value. Fig. 4.23 plots the I_{on} and I_{off} of each DUT measured by the proposed test structures with and without applying the current-cancellation technique. As shown in Fig. 4.23, the I_{off} measured with current cancellation applied ranges over a wide interval (from -8 to -4 A.U.). However, the I_{off} measured without current cancellation is uniformly high, in a small interval (from -5 to -4 A.U.). This result shows that if the current-cancellation technique is not employed, the leakage current from the transmission gates may dominate the I_{off} measurement and hence the true I_{off} distribution cannot be measured .

4.3.4 Proposed Array vs. ROM-like Array

In the following experiment, we fabricate the proposed array-based test structure next to a ROM-like array-based test structure, using a mature, relatively old process technology. Both test structures can be placed into a standard scribe line for monitoring process variation. Similar to Fig. 4.16, we choose two nearby DUTs for measurement, one from each test structure. Fig. 4.24 plots the I_d versus V_g of the chosen DUT measured by each test structure. Compared to our proposed test structure, the I_d measured by the ROM-like test structure is larger, and hence its

measured V_{th} is smaller. This difference results from the larger leakage current of the common-gate and common-drain buses in the ROM-like transistor array, which shows the potential measurement error from modifying a DUT's layout for characterization of process-variation.

Next, we fabricate these two array-based test structures in an advanced, newly developed process technology. Fig. 4.25 plots the I_{on} and V_{th} measured on each DUT of both test structures. The median and variance of V_{th} measured by both test structures are listed in the upper-right corner of Fig. 4.25. Compared to the proposed test structure, the V_{th} measured by the ROM-like test structure is approximately 10% smaller, which is consistent with the result shown in Fig. 4.24. Also, the V_{th} variation measured by the ROM-like test structure is about 12% smaller. This smaller V_{th} variation results from the fact that the long straight poly line gate busses used in the ROM-like transistor array do not suffer the shortening and rounding effects which occur in the short poly gates of both the proposed transistor array and typical real product circuits. Therefore, the V_{th} variation measured from a ROM-like test structure may be unrealistically smaller than the variation which occurs in a real circuit.

4.3.5 Hardware IR compensation v.s Adaptive Voltage compensation

Hardware IR compensation is one key technique to ensure the measurement accuracy of an array-based test structure. In this subsection, we compare the proposed hardware IR compensation with the adaptive voltage compensation introduced in Section 5.1.2. Table 5.1 summarizes the comparison between the two IR compensation schemes. First, the hardware IR compensation requires half as many SMUs per DUT compared to adaptive voltage compensation. Second, the measurement accuracy of hardware IR compensation can be calculated by Equation 4.2 and

further controlled by modifying the internal resistance of the SMUs. The measurement accuracy of the adaptive voltage compensation is controlled by the size of the sweep step. A smaller sweep step can increase the measurement accuracy, but only at the cost of increased test time. Next, hardware IR compensation requires no sweeping of the forcing voltage and hence results in a faster convergence time. Last, the test time for measuring a DUT with hardware IR compensation is around 10ms, which is 50x smaller than that with adaptive voltage compensation. This short test time and high measurement accuracy further demonstrates the efficiency and effectiveness of the proposed hardware IR-compensation scheme. To illustrate the impact of this reduction of test time, we note that to obtain the 256 data points for each data set in Fig. 21, 128 seconds are required for measurement using adaptive voltage compensation, as compared to only 2.56 seconds using hardware IR compensation.

	adaptive voltage compensation	hardware IR compensation
# of SMU required	8 (each MOSFET)	4 (each MOSFET)
Accuracy	depends on sweep step	>99.991%
Convergence Time	slow	fast
Testing Time	~500msec	~10msec

Table 4.1: Comparison between hardware IR compensation and adaptive voltage compensation.

4.3.6 Local Mismatch Measured by Proposed Test Structure

Accurate matching of active and passive devices is critical for analog and mixed-signal circuits, which usually require a high level of precision. As process variation becomes larger in advanced process technologies, the degree of mismatch on devices actually determines and limits the performance of analog and mixed-signal circuits [63] [64] [65]. To measure mismatch in a process technology, two nominally identical MOSFETs directly adjacent to each other are fabricated and then mea-

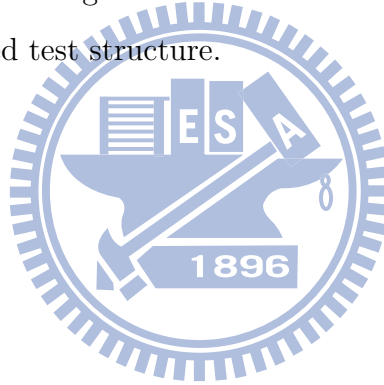
sured. The physical difference between these two nominally identical MOSFETs is defined as *local mismatch*. In this subsection, we utilize the proposed array-based test structure to collect mismatch data for a large number of pairs of adjacent MOSFETs.

In order to place a pair of identical MOSFETs into a test unit shown in Fig. 4.8, the number of DUT terminals connected to pads must be less than or equal to four. Fig. 4.26 shows the DUT design used in our array-based test structure for measuring local mismatch. As Fig. 4.26 shows, the two MOSFETs are symmetric and share a common gate and source. Also, the source and bulk of each MOSFET are tied together to save another terminal connection to a pad. When measuring the V_{th} of the left (right) MOSFET, we float Drain-2 (Drain-1), apply voltages to the common gate in a sweep step, ground the common source/bulk, and sense the current at Drain-1 (Drain-2).

In the following experiment, we fabricate the proposed array-based test structures to measure local V_{th} mismatch on an advanced process technology. Each test structure contains 256 paired MOSFETs, which include the following four device dimensions (listed in order of decreasing channel area): long-channel devices, standard-cell devices, on-rule devices and sub-rule devices. For each dimension, 64 paired MOSFETs are included. Fig. 4.27 shows the difference of the normalized V_{th} between each of the paired MOSFETs for each dimension. The V_{th} mismatch significantly increases when the channel area decreases, consistent with the theoretical behavior of random dopant fluctuation [66].

V_{th} mismatch also depends strongly on the difference in the poly gate CD (critical dimension) between the two devices. We use an in-line SEM (scanning electron microscope) to physically measure the poly CD difference between each of the paired MOSFETs. Fig. 4.28 plots this physically measured mismatch of poly CD

for the same group of paired MOSFETs shown in Fig. 4.27. Fig. 4.29 in turn plots V_{th} vs poly CD for all MOSFETs in the group of MOSFET pairs. The physical poly-CD mismatch shown in Fig. 4.28 and the spread of the V_{th} populations in Fig. 4.29 indeed correlates with the electrical V_{th} mismatch shown in Fig. 4.27, which confirms that local poly CD difference is one of the sources of this V_{th} mismatch. The measurement results in Fig. 4.27 and Fig. 4.28 demonstrate that the mismatch between paired MOSFETs may significantly vary within a close proximity (note that the distance between two DUTs, i.e. two MOSFET pairs, in the proposed test structure is 30um). In addition, the strong correlation between our electrical and physical measurements again demonstrates the accuracy and effectiveness of the proposed array-based test structure.



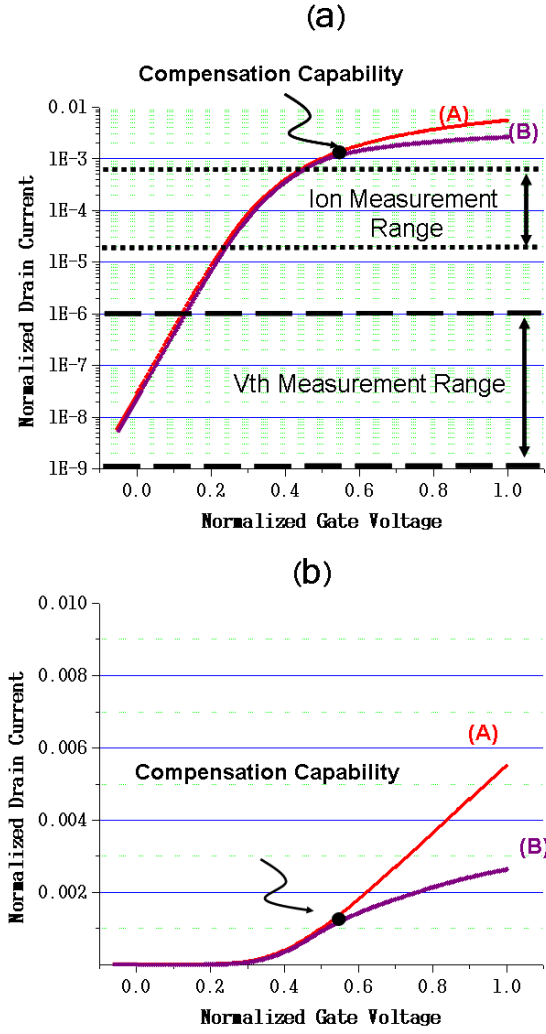


Figure 4.16: I_d versus V_g measured by (A) a PCM testline and (B) a proposed array-based test structure with log scale (upper figure) and linear scale (lower figure).

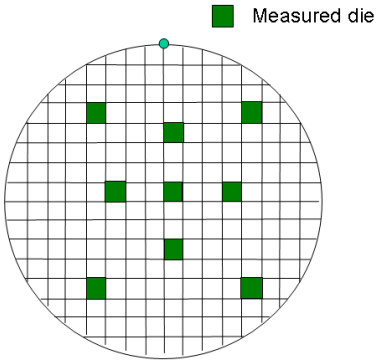


Figure 4.17: Locations of measured test structures.

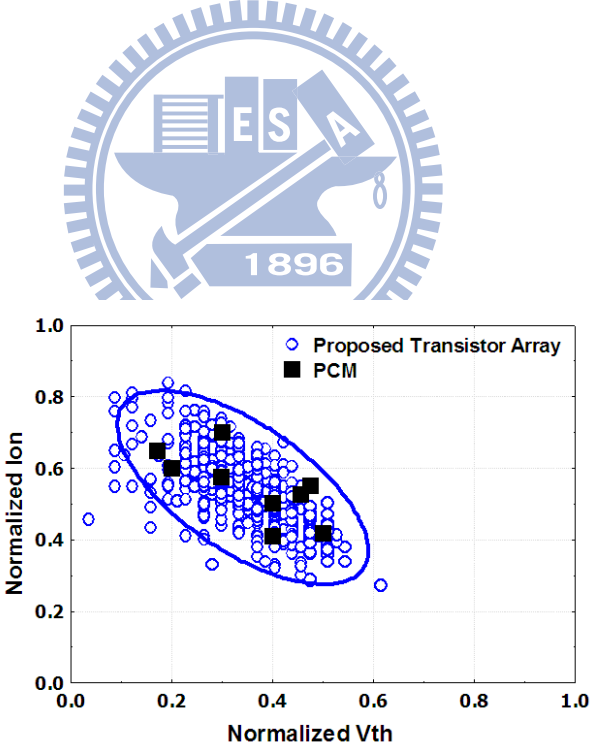


Figure 4.18: I_{on} and V_{th} of each DUT measured by 9 PCM testlines and 9 proposed test structures. The curve is an elliptical fit of the 95% confidence interval of the data set.

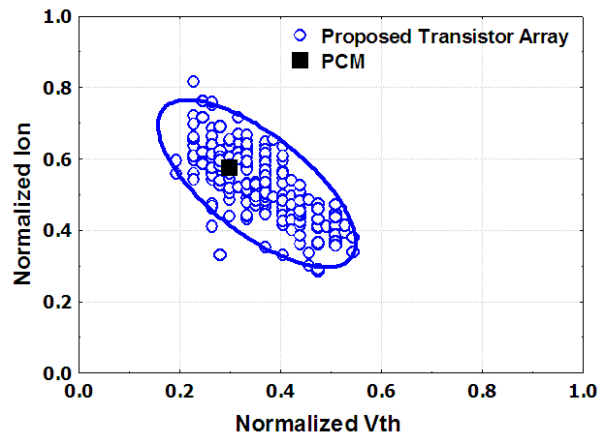


Figure 4.19: I_{on} and V_{th} of each DUT measured by only one PCM testline and one proposed test structure. The curve is an elliptical fit of the 95% confidence interval of the data set.

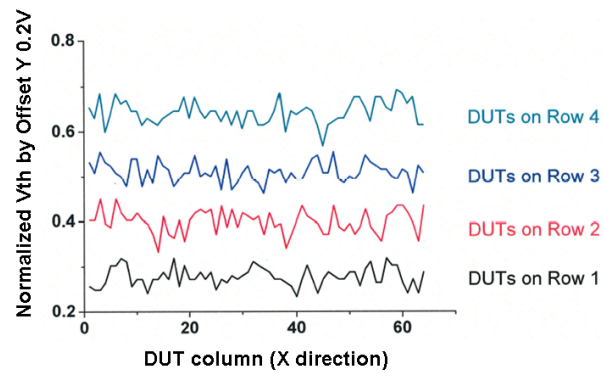


Figure 4.20: V_{th} vs DUT column (X axis) for all four rows of the structure, with each row shifted by a fixed offset along the Y axis.

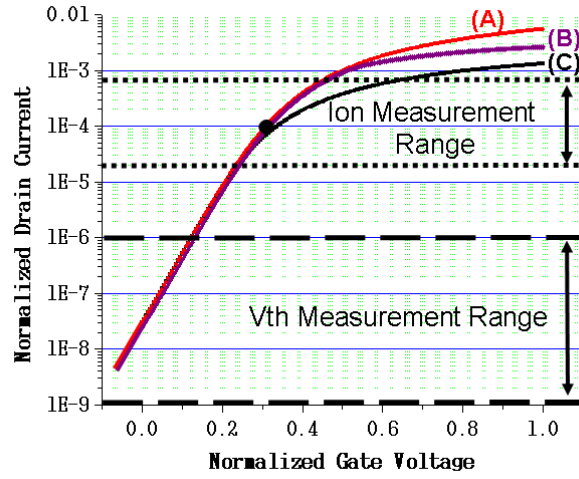


Figure 4.21: I_d versus V_g measured by (A) a PCM testline, (B) a proposed array-based test structure and (C) a proposed array-based test structure without applying hardware IR compensation.

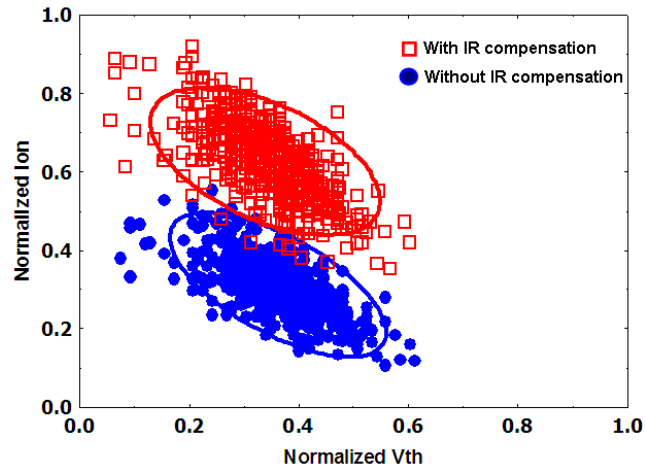


Figure 4.22: I_{on} and V_{th} of each DUT measured by proposed test structures with and without the hardware IR compensation. The curves are elliptical fits of the 95% confidence intervals of the two data sets.

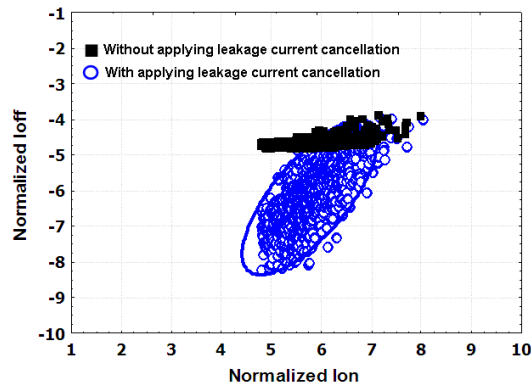


Figure 4.23: I_{on} vs. I_{off} of each DUT measured by the proposed test structures with and without applying the current-cancellation technique.

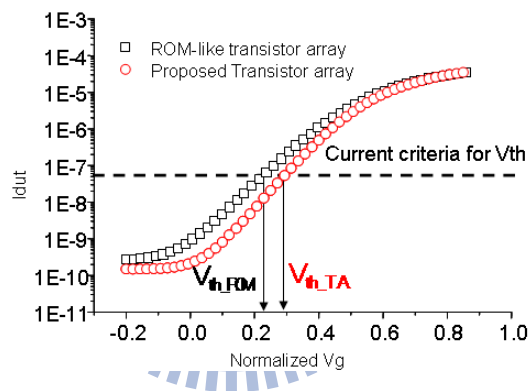


Figure 4.24: I_d versus V_g measured by a ROM-like DUT array and a proposed DUT array.

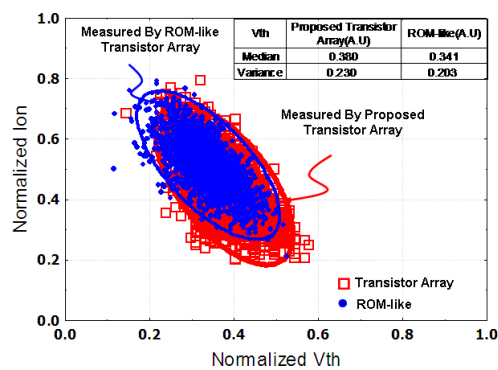


Figure 4.25: I_{on} vs. V_{th} of each DUT measured by a ROM-like DUT array and a proposed DUT array.

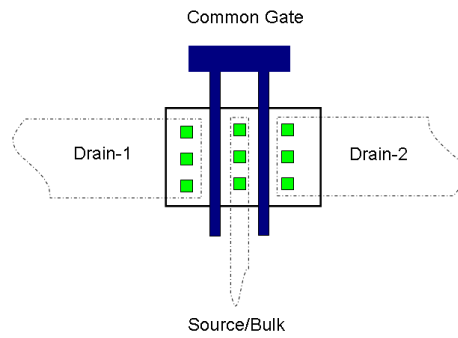


Figure 4.26: Layout of a paired-transistor DUT.

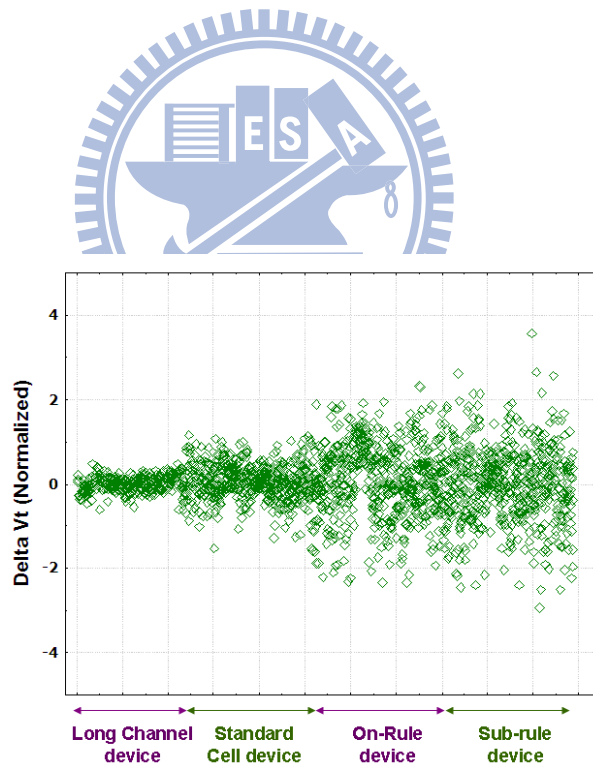


Figure 4.27: V_{th} mismatch of paired MOSFETs measured by the proposed test structures for different W/L dimensions.

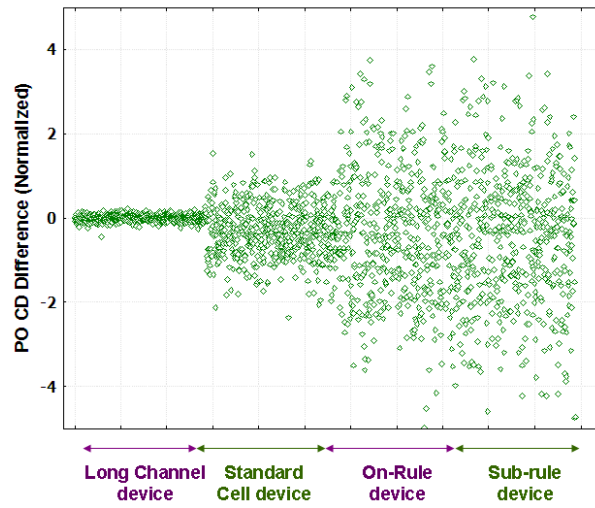


Figure 4.28: Poly-CD mismatch of paired MOSFETs measured by scanning electron microscope for different W/L dimensions.

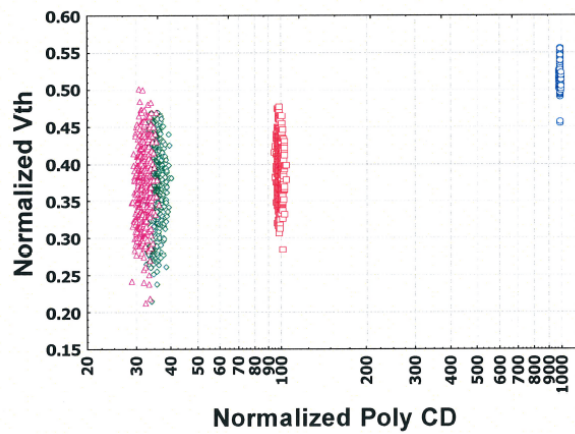


Figure 4.29: V_{th} vs poly CD for all MOSFETs in the array of MOSFET pairs.

Chapter 5

Fast transistor threshold voltage measurement method for high-speed, high-accuracy advanced process characterization

As the feature size of devices scales down, the device variability imposed by each process step does not scale accordingly. As a result, the process variation of advanced process technology nodes has greatly increased and has become a critical factor in both IC design and manufacturing [67]. In order to design and manufacture in the presence of process variation, much research effort has focused on the areas of measurement, analysis, and modeling of variation during the past decade [68] [69] [70] [71] [72] [73] [74] [75] [78]. Furthermore, modeling and design for manufacturing (DFM) of increasingly complex process technologies incorporating process features such as stressed contact etch-stop layers, SiGe S/D [75], SMT [77], etc., requires a much larger range of test structures and larger data volume in order to accurately characterize the layout-dependent effects (LDE) resulting from these process features. The need to accurately characterize both process variation and increasing complex LDE has dramatically increased the amount of testing required during all stages of process development and manufacturing, which in turn demands the development of more efficient test structures and methods which can provide this data without increasing test time to unfeasible levels. During conventional device characterization for the study of LDE and process variation, a conventional test

structure, i.e. a *PCM (process control monitor) testline*, is placed in a wafer's scribe line. The PCM testline has DUTs and IO pads aligned along a straight line and uses four IO pads to measure each DUT. Thus, both the height of a PCM testline and the required spacing in a scribe line are both limited [84]. Only a relatively limited number of DUTs can be placed in such a testline configuration due to the limited number of IO pads. In order to effectively utilize limited scribe-line space to place and individually measure a sufficiently large number of DUTs to address the characterization needs of DFM, LDE, and process variation, several array-based test structures have been proposed to share IO pads among DUTs and hence reduce the number of IO pads required between the DUTs [81] [82] [79].

Transistor threshold voltage (V_t) is a commonly used parameter to quantify transistor performance both during process development and volume manufacturing [85]. There are several different definitions of the threshold voltage of a MOSFET device [89]. The most commonly used definition for process monitoring in IC foundries is *constant current V_t* . The constant current V_t measurement is generally performed by a binary search algorithm. However, the binary search algorithm requires much longer testing time compared with measurement of other MOSFET parameters such as I_{on} , I_{off} etc. Moreover, the number of DUTs significantly increases when array-based test structures are employed to collect a large enough sample size for statistically meaningful results, and the V_t measurements represent the most time-consuming portion of the characterization of these DUTs.

Operational amplifier (op-amp or OP) based methods have been proposed to simplify and accelerate V_t measurement [86]. Recently, the work of Ref. [88] has also proposed similar methods, employing an on-chip op-amp design combined with an addressable FET array. Such op-amp schemes enable rapid characterization of V_t distributions with large numbers of data samples. However, there are several

major challenges in on-chip OP-based amplifier design for V_t measurement. First, the V_t measured by OP-based test structures may be impacted by body effect due to non-zero source voltage, e.g. as would occur in the circuit of [88]. Second, variation of on-chip load resistance (R_{load}) can result in an inaccurate V_t measurement. This latter issue will become much more severe in coming advanced process technology nodes due to the scaling of feature size without corresponding scaling of variability. Lastly, schemes utilizing on-chip OP-based structures are limited by the op-amp accuracy and gain. In 0.35 μ m technology, an OP gain of 100dB is easily achieved using folded cascode design. However, the gain significantly decreases in advanced technologies due to small transistor output resistance and reduction in headroom. In such advanced technologies, the design of op-amps with gain as high as 100dB requires additional circuitry. As a result, the layout area devoted to the op-amp significantly increases, which might prevent the design of practical testlines which can be placed within the wafer scribe line. Furthermore, in general, device mismatch is very poor in the early stages of process development. Therefore, there may be substantial errors in the V_t measured by on-chip OP-based test structures during the early stages of process development.

In this paper, we propose a design and methodology for V_t measurement using high gain and high accuracy operational amplifier-based SMUs directly connected to the DUT. The OP-based SMUs are implemented using discrete ICs which are well calibrated by the tester supplier, and thus are not sensitive to process variation. The experimental results which we present, based on advanced process technologies, demonstrate that the proposed design reduced V_t testing time by a factor of 6 to 10, while simultaneously delivering improved accuracy, with a V_t standard deviation below 0.15 mV. Moreover, combined with array-based test structure, the test time can be further improved due to time overhead saving from the connect,

disconnect operation between SMUs and testlines IO pads and prober index time, which is prober chuck moving time from one testline to another.

5.1 Background

5.1.1 Measuring constant current V_t using a binary search algorithm

In a traditional parametric tester V_t measurement, the SMUs are configured as voltage sources which can be modeled as unity gain buffers as shown in Fig. 5.1 [84]. Each IO pad is connected to the forcing and sensing node of a SMU through the probe card during testline measurement. In this architecture, the voltage at the connection of the forcing and sensing current paths, commonly referred to as the *compensation point*, must be equal to V_{set} since no current flows through the sense path. Therefore, the IR drop caused by parasitic resistance from the tester to the compensation point is completely eliminated. For the V_t measurement, the drain, source, gate and bulk terminals are connected to this unity-gain SMU as shown in Fig. 5.2. In the constant-current method of V_t measurement, V_t is defined as the gate voltage resulting in the flow of a specific drain current. [85], [86]. For example, V_t can be defined as the gate-to-source potential required to drive the threshold drain-to-source current, $I_{ds}(V_t) = (I_{0,n}) \cdot W_{eff} / L_{eff}$ for n-FET, and $I_{ds}(V_t) = (I_{0,p}) \cdot W_{eff} / L_{eff}$ for p-FET, where $I_{0,n}$ and $I_{0,p}$ are parameters of a given process technology for n-FET and p-FET, respectively. In this paper, we use $I_0 = 20\text{nA}$ for both N and PMOS. Before beginning the binary search for V_t measurement, the following parameters of the binary search must be specified: (a) gate voltages V_{gstart} and V_{gstop} , which specify the range of gate voltages to be searched, and (b) a convergence criteria for matching the target current. In the first iteration, SMU2 forces the search voltage value, $V_{force} = (V_{gstart} + V_{gstop}) / 2$, to the gate node of the DUT while SMU1 measures the drain current (I_{ds}) and compares it to the target value.

If the matching criteria $(I_{ds} - I_{target})/I_{target} < \text{matching tolerance}$, is met, V_t is assigned to V_{force} . However, if it is larger than the matching criteria, another iteration must be performed. Before proceeding to the next iteration, V_{force} must be modified. If $I_{ds} > I_{target}$ then SMU2 forces the search voltage value $V_{force} = (V_{gstart} + V_{force})/2$. In contract, if $I_{ds} < I_{target}$ then SMU2 forces the search voltage value $V_{force} = (V_{force} + V_{gstop})/2$.

In subsequent iterations, the SMU2 applied gate voltage values above and below V_{th} , which become increasingly close to V_{th} with successive iterations until the current measured by the sense unit matches the target value within the specified criteria. Fig. 5.3 lists the pseudo-code implementation of a binary search algorithm for finding the gate voltage resulting in a specified drain current. The accuracy of the result obtained by this algorithm strongly depends on the convergence condition and the maximum number of iterations, which thus presents a trade-off between testing time and accuracy. Fig. 5.4 schematically illustrates the iteration of forcing voltage and measuring current for V_t measurement using the binary search algorithm.

Therefore, the binary search algorithm for V_t measurement typically requires much longer testing time than the measurement of other device parametrics such as I_{on} , I_{off} and subthreshold swing. For the measurement of a number of DUTs large enough to obtain a statistically significant quantification of process variation, or to perform adequate characterization for LDE model or DFM verification, the test time correspondingly increases and may limit the number of DUTs which can be measured, thus also limiting the accuracy of the characterization.

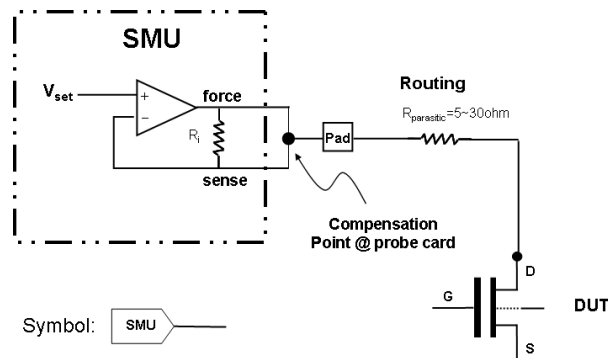


Figure 5.1: Voltage compensation mechanism used in a conventional PCM testline.

5.1.2 Improving constant current V_t testing time using an interpolation methodology

One approach to reduce the number of SMU force-measure iterations below the number required by the binary search algorithm is to apply an interpolation algorithm. The number of force-measure iterations can be reduced from 6–9 iterations for the binary search algorithm to 2 for the interpolation algorithm. The interpolation method is performed by setting the two initially defined gate voltages, V_{g_hi} and V_{g_lo} , to values corresponding to drain current values very close to the target current (e.g. $I_{0,n} \cdot W_{eff}/L_{eff}$ for n-FET). To minimize the interpolation error, one gate voltage is set to drive a drain current slightly higher than the target current, and the other gate voltage drives a drain current slightly lower than the target, as illustrated in Fig 5.5. The V_t value can be simply obtained by interpolation due to the linear I-V characteristics in the subthreshold region. This interpolation methodology is easily applied to measuring the variation of DUTs with identical or nearly identical dimensions because the same initial voltage settings and current criterion can be used for all DUTs. However, for the characterization of many DUTs with different transistor dimensions, e.g. for DFM, SPICE modeling or other process characterization, it may be difficult to define initial gate voltages appropriate to all

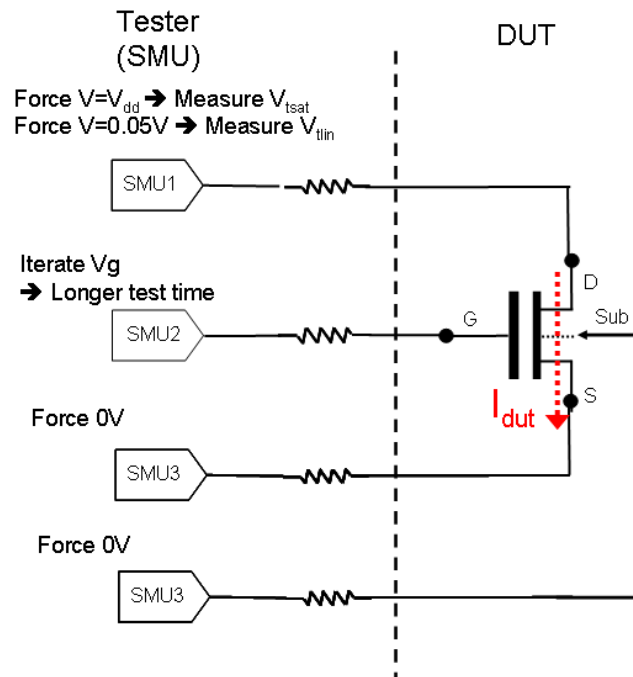


Figure 5.2: SMU connection and bias condition for V_t measurement using a binary search approach for n-FET.

DUTs. Moreover, in the case of a MOSFET fabricated using shallow-trench-isolation (STI), the gate region runs flat across the isolation. Consequently, the portion of the gate over the field region creates a two-dimensional fringing field on the corner and sidewall region as shown in Fig. 5.6(a), which effectively creates a corner parasitic MOSFET in parallel with the main DUT. The parasitic device turns on at a gate voltage lower than that of the main DUT channel, resulting in a "hump" in the I_d - V_g curve as shown in Fig. 5.6(b) [87]. As illustrated in Fig. 5.6(c), a MOSFET with such a hump in its I_d - V_g characteristic exhibits a significant difference between the values of V_t obtained by interpolation and binary search methodologies respectively. Therefore, this interpolation technique cannot be applied to devices exhibiting non-linear I-V characteristics in the subthreshold region, such as those with parasitic corner devices resulting from STI. Since STI is a common feature of almost

```

1.  double Va=Vgstart;;
2.  double Vb=Vgstop;
3.  double Itarget=It0*Weff/Leff
4.  double converge_criteria=1%;

5.  connect(smu1,drain);
6.  connect(smu2,gate);
7.  connect(smu3,source);
8.  connect(smu4,bulk);
9.  Vforce=(Va+Vb)/2;
10. for(int i=0; i<N; i++)
11. {
12.     force_v(gate_pin,Vforce);
13.     measure_i(drain_pin,Ids)
14.     double match=(Ids-Itarget)/Itarget*100;
15.     If(match < converge_criteria)
16.     {
17.         Vth=Vforce;
18.         return;
19.     }
20.     else
21.     {
22.         if(Ids<Itarget)
23.         {
24.             Va=Vforce;
25.             Vforce=(Vb+Vforce)/2;
26.         }
27.         If(Ids>Itarget)
28.         {
29.             Vb=Vforce;
30.             Vforce=(Va+Vforce)/2
31.         }
32.     }
33.     Vth=Vforce;
34. }

```

Figure 5.3: Pseudo code of binary search V_t measurement.

all modern CMOS technologies, this severely limits the utility of the interpolation technique.

5.1.3 Fast V_t measurement using an on-chip operational amplifier based test structure

Although the number of force-measure iterations can be reduced from ~ 9 for binary search to 2 for the interpolation algorithm as described above, the V_t value obtained by the interpolation algorithm is less accurate than that obtained by binary search. The number of force-measure iterations for constant current V_t measurement can actually be further reduced to only a single iteration, while maintaining higher measurement accuracy, by adopting op-amp-based test structures utilizing an on-chip op-amp. This technique for V_t measurement using only one force-measure iteration is illustrated in Fig. 5.7. Fig. 5.8 shows the circuit schematics for the op-

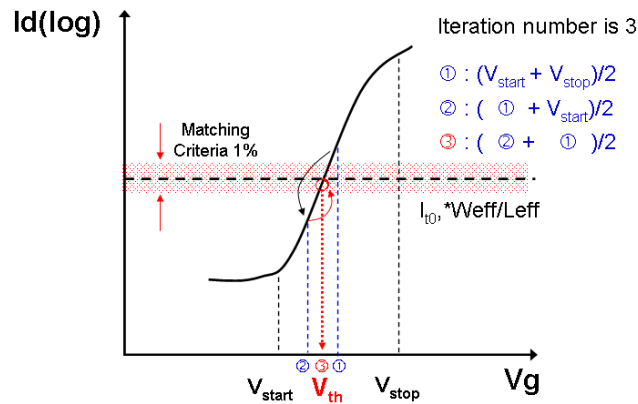


Figure 5.4: Example of iterating V_g to obtain drain current matching target within specified criteria.

amp based test structure for V_t measurement, where the DUT is (a) n-FET, and (b) p-FET, respectively. In Fig. 5.8(a), V_{set} , V_{ss} , V_b and V_d are input terminals and V_g is the op-amp output, which is connected to both the DUT gate and a digital voltmeter (DVM) which in turn measures V_g [86], [88].

In Fig. 5.8 (a), where the DUT is a NFET, the DUT source voltage is forced to V_{set} by the op-amp feedback loop. The higher accuracy V_t measurement is performed by forcing a precise bias current of $(V_{set}-V_{ss})/R_{load}$, where R_{load} is the resistance of a precision load resistor and V_{ss} is typically equal to 0V for the characterization of a NFET. By appropriate selection of V_{set} , this current is set equal to the threshold current value and is fed to the source terminal of the DUT. The OP output voltage is automatically modulated by the OP feedback loop to maintain the threshold current, and it quickly converges to a voltage equal to the V_t of the DUT. Fig. 5.8(b) shows the V_t measurement setup for a p-FET, which is similar to that for a n-FET except that $V_{set}=V_{dd}$.

However, there are several potential issues which may arise in this configuration. First, V_t measured by OP-based test structures may be impacted by body

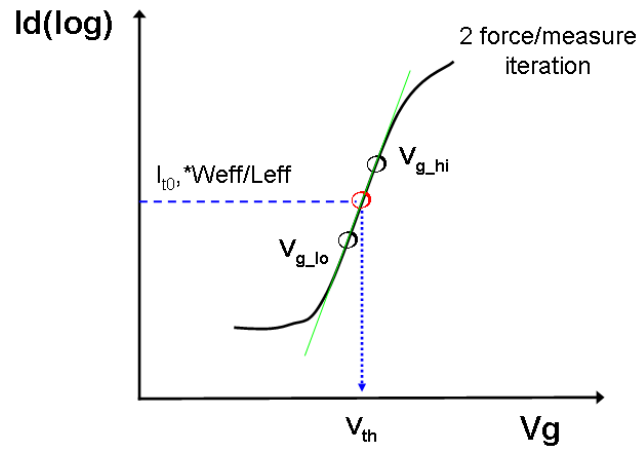


Figure 5.5: Setting the initial gate voltage values to interpolate to the gate voltage corresponding to the target drain current. Log scale is used only for ease of visualization. In practice, linear scale is used for V_t interpolation.

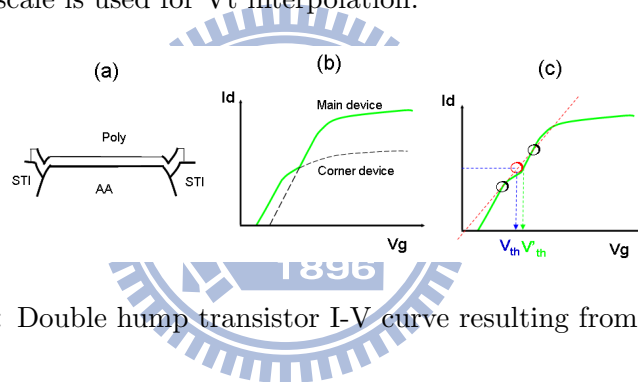


Figure 5.6: Double hump transistor I-V curve resulting from use of STI.

effect due to non-zero source voltage. For example, in the case of the NFET, if V_{ss} is grounded, then the bias current results in a small positive voltage at the DUT source terminal. If the DUT body is grounded, this results in body effect due to the positive voltage difference from source to body. Although, the impact of body effect may be less severe in silicon-on-insulator (SOI) technologies such as those studied in Ref. [88], the impact may not be negligible in bulk silicon technology, which is far more commonly used. Second, in general, the DUT test structure utilized for V_t measurement must also be capable of measuring I_{on} , I_{off} and similar device parameters. The circuit configuration of Ref. [88] can likely only be employed for V_t measurement and will be unable to measure other parameters. Third, on-chip load

resistance (R_{load}) variation can greatly compromise the accuracy of the V_t measurement, an issue which will become more severe in forthcoming advanced process technology nodes. Further, such a test circuit relies on an on-chip precision resistor, but the value of the precision resistor may also not be on target until the later stages of process development. The test program for this test structure must first characterize the precision resistor and then must adjust bias voltages accordingly. The fourth and final issue is the accuracy of the on-chip OP. In 0.35 μ m technology, OP gain of 100dB using a folded cascode OP is readily achieved, but in advanced technologies the low transistor output resistance and the aggressive power supply scaling from 3.5V to below 1.5V result in reduced headroom and thus significantly reduced OP gain. In addition, device mismatch is generally not well controlled in the early stages of process development. Therefore, the V_t measured by test structures utilizing on-chip OPs can easily fail to accurately measure the true threshold voltage.

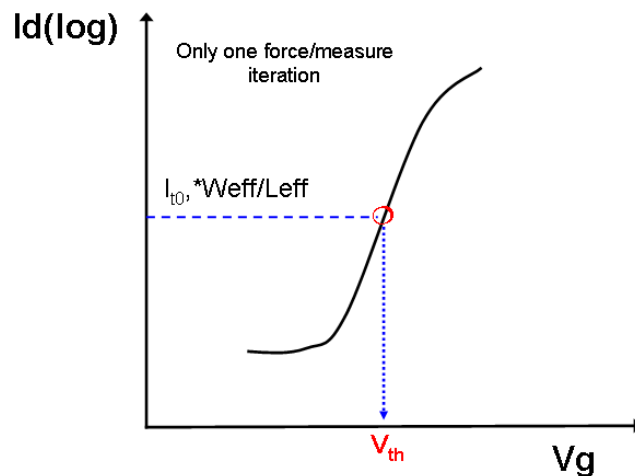


Figure 5.7: Only a single force-measure iteration is required by the OP-based V_t measurement technique.

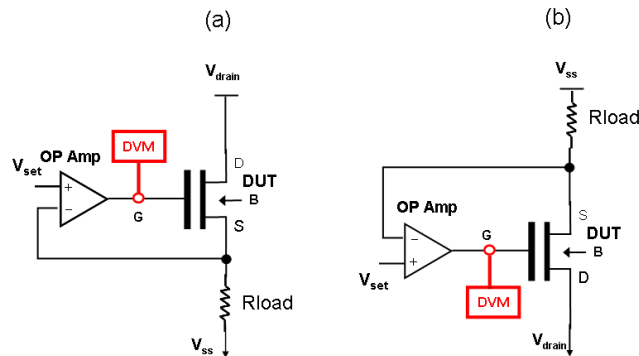


Figure 5.8: Circuit schematics for OP-based V_t measurement, where the DUT is (a) n-FET, and (b) p-FET.

5.2 Design Methodology

5.2.1 OP-based SMU for Fast V_t Measurement

As discussed in section 5.1.3 above, the reduced headroom in advanced process technologies might result in on-chip op-amp gain which is too low to achieve reliable V_t measurement. In addition, the voltage gain of an on-chip op-amp may significantly vary with transistor mismatch due to process variation, especially in advanced technologies below 65nm. Rather than using an on-chip OP design, for the novel fast V_t measurement technique presented in this paper, we modify the configuration of the OP in the tester's SMU. Instead of the conventional unity-gain buffer configuration (e.g. as shown in Fig. 5.1), the SMU OP is connected as shown in Fig. 5.9 by separating the force and sense nodes and reconnecting the sense node to the DUT source node. In such a configuration, in contrast to the circuits of Fig. 5.1, SMU2 is no longer functioning as a unity-gain buffer at the gate terminal. Instead, in the proposed configuration, it is configured as a two stage operation amplifier, ie. SMU2 plus DUT are connected to form a voltage follower. In addition, since the tester's SMU op-amp is constructed using discrete ICs, it can easily achieve very high-gain operation while remaining free from sensitivity to process variation.

The overall gain of this two stage OP exceeds 100dB. Therefore, the inverting input node and the non-inverting input can be considered connected by a virtual short. For the V_t measurement, V_{set} is 0V, and the voltage at the source terminal is also forced to almost exactly 0V due to the following two following two reasons. First, the op-amp's gain is high enough to enable the op-amp to effectively maintain the virtual short between the two op-amp input nodes. Second, the current ($< 0.1\mu A$) flowing between sense pad and source node is small enough that the IR drop in this path has negligible impact. Meanwhile, SMU3 functions as the current source, forcing the negative target current used in the constant-current V_t definition, which flows completely through the DUT due to the high input impedance of the op-amp input terminals. Therefore, the output node of the op-amp is able to quickly drive the gate voltage to the correct value of V_t once V_{set} , V_d , V_b and the target current for the constant current V_t definition have been assigned. V_t can be measured in either the saturation or the linear region by appropriately adjusting V_d . Typically, $V_d = V_{dd}$ and $V_d = 0.05V$ are the bias conditions for saturation-mode V_t (V_{ts}) and linear-mode V_t (V_{tl}), respectively. This configuration affords several advantages. First, this configuration has high V_t measurement accuracy. The accuracy is better than the binary search approach because there is almost no error in the target current which is forced by a second SMU. The second benefit of this OP-based SMU approach is the absence of body effect in V_t measurement. The third one benefit is that multiple additional device parametrics, such as I_{on} , I_{off} , etc., can be measured in this configuration. For the V_t measurement, note that we deliberately avoid connecting the current source to the DUT drain terminal, because a less accurate V_t value would result from connection to the drain due to an extra junction leakage current path which would remove part of the reference current $I_{ds}(V_t) = I_{0,n} * W_{eff} / L_{eff}$.

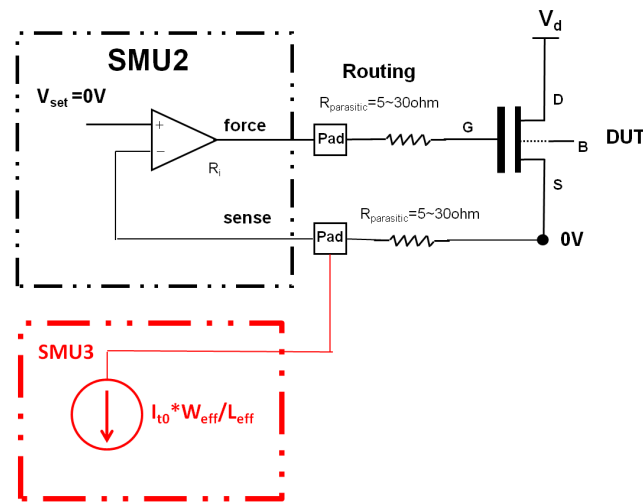


Figure 5.9: The configuration of an OP-based SMU and a n-FET for V_t measurement with one force-measure iteration. The target current defined for V_t is forced as a negative current by an additional SMU.

5.2.2 Implementation for Stand-alone DUT

Fig. 5.10 illustrates the SMU connections for fast V_t testing utilizing an OP-based SMU. Other than the OP-based SMU2 connected to the gate and source terminals, there are two unity-gain buffers for the drain and bulk terminals, and one current source for the source terminal. SMU2, configured with separated force and sense nodes, is connected to both the gate and source terminals of the DUT. The drain and bulk terminals are connected to SMU1 and SMU4, respectively, and the negative current (i.e target current) source is provided by SMU3. The body effect issue faced by on-chip OP test structures is completely eliminated in this test configuration. Four SMUs are required for the fast V_t measurement, the same number required by the binary search algorithm. Since the OP-based SMU can read out the output voltage of the OP, there is no need for an additional SMU to sense the voltage at the gate terminal as was required in the circuit of Fig. 5.8 . Fig. 5.11 lists the fast V_t algorithm to obtain V_t with only a single force-measure iteration.

Note that the voltages at each terminal must be forced in the sequence shown, i.e. first drain, then bulk, then gate, and lastly the negative current should be forced at the source. If this forcing sequence is not strictly observed, the voltage overshoot might damage or breakdown the DUT.

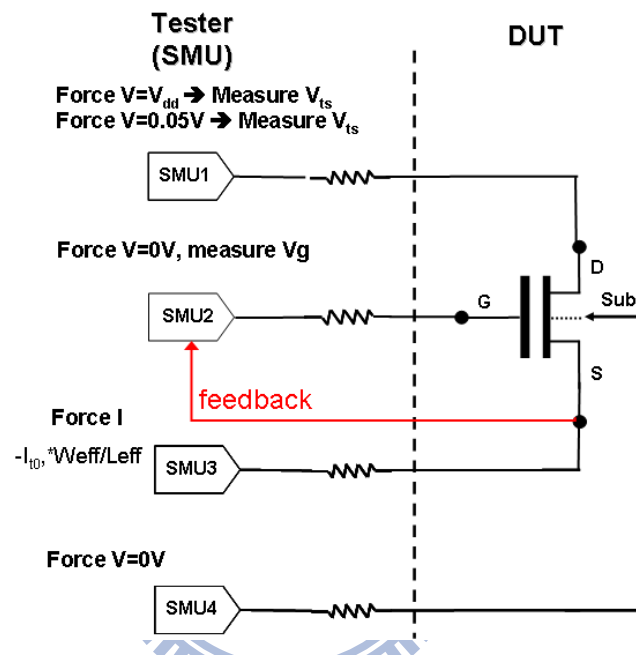


Figure 5.10: SMU connections for fast V_t measurement by using the proposed OP-based SMU .

```

1. connect(smu1,drain);
2. connect(smu2,gate,source);
3. connect(smu3,source);
4. connect (smu4,bulk);

5. Itarget=It0*Weff/Leff

6. Force_V(Drain, Vd);
7. Force_V(Bulk, Vb);
8. Force_V(Gate, 0V);
9. Force_I(Source, Itarget)
10. Measure_V(Gate, Vth)

```

Figure 5.11: Pseudo code of V_t measurement by OP-based SMU.

5.2.3 Implementation for Array Test Structure V_t measurement

An addressable array containing multiple transistors with various values of width and length (W/L) is used for the characterization of V_t variation. Fig. 5.12 shows the proposed transistor array with $16 \times 64 = 1024$ test units. Each test unit consists of a few transmission gates and one DUT (a FET in this example). The DUT can be measured by selecting the corresponding test unit through the column decoder. As shown in Fig. 5.12, the gate terminals of the FETs in a selected column are connected to SMU2 and the source sense terminals are connected to SMU3, which is fed back to SMU2, while the switches connecting drain, source, gate, and bulk terminals in unselected columns are turned off. Typically, the transmission gates need to be sized large enough to have negligible voltage drop at the current level required for V_t measurement. However, in this experiment the drain and source terminals of all DUTs are connected to one SMU with force/sense IR drop compensation, respectively. With this voltage compensation mechanism, wide metal routing for the drain and bulk terminals is not required to reduce the parasitic resistance of these connections, allowing a more compact testline layout. The V_t of the selected DUT is read out by SMU2. In addition, all periphery circuits, such as latches and decoders, are designed with 2.5V I/O devices so that their background leakage current, including sub-threshold leakage current and gate-oxide leakage current, can be reduced, and their performance will not be affected by any process variation which may be present in the advanced process under study.

5.3 Experimental Results

5.3.1 Binary search V_t testing time

The time required for a single binary search force-measure iteration consists mainly of contributions from two stages in the execution of the algorithm: first, the

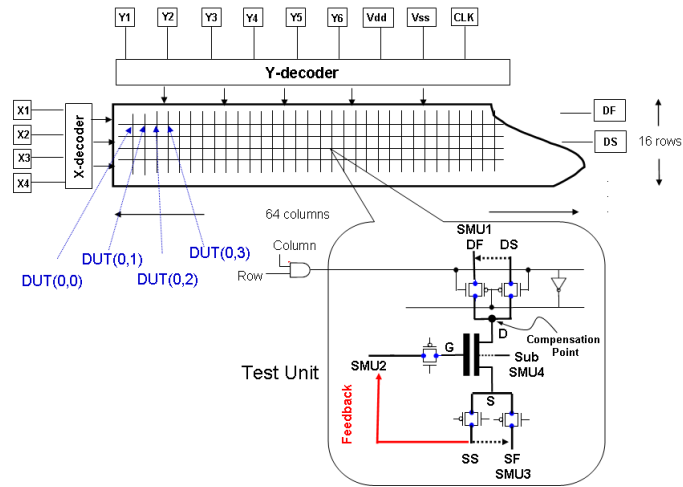


Figure 5.12: V_t measurement by OP-based SMU in an Array Test Structure.

forcing of the voltage, and second, the current measurement. The time required to force a voltage is principally a function of the SMU settling time, which is relatively short, typically ~ 100 μ s. The time required for current measurement is determined by the current level to be measured and the sample size, which is defined by the integration time. This second contribution is strongly dependent on the magnitude of the target current if the integration time is set to the same value. For measurement of a lower current level, the SMU requires a longer measurement time because it must switch from the measurement mode for a higher current to that of a lower current. Typically, the time required for changing the current range a few ms because it requires the switching of electro-mechanical components in the tester.

In this experiment, it was determined to require approximately 7 ms and 30 ms for one force-measure iteration at the μ A and nA current levels, respectively. As discussed in section 5.1.1, V_t is defined by the gate voltage resulting in the measured drain current matching the target current $I_0 \cdot W_{eff} / L_{eff}$ within a specified criteria, for instance 1%, which introduces about 0.5mV error based on SPICE simulation. Fig. 5.13 curve A plots the gate voltage (V_g) vs. iteration number throughout

the successive force-measurement cycles of a V_t measurement by binary search. During the course of the measurement, V_g is set successively to values above and below V_t , approaching V_t with an increasingly tight tolerance, and converging when the drain current I_d (measured by SMU1 in the connection scheme of Fig. 5.2) approximately matches the target current. Curve B plots the percent mismatch between the measured drain current and the target current, showing that typically 9 or 10 iterations are required to reach the 1% matching criteria.

In order to obtain representative V_t measurement times for the binary search algorithm, transistors of two different channel lengths were evaluated. Transistor A is a longer channel device with a target current in the nA range. Transistor B is a short channel device with a target current of order 1 μ A. Table 5.1 summarizes the device characteristics and V_t testing times obtained using the binary search algorithm. The shorter channel transistor with μ A target current required approximately 69 ms, whereas the longer channel transistor required a testing time of approximately 270 ms. The number of iterations required for measurement of the longer channel transistor is slightly higher than for the shorter channel transistor due to the lower target current to be measured. Moreover, the testing time for the lower target current significantly increases by nearly 4X due to the additional time required for the SMU range to switch in order to accommodate the lower current level.

	Transistor A	Transistor B
device size	longer channel	short channel
target current	\sim nA	\sim μ A
average of testing time	270.06msec	68.75msec
std. dev of testing time	8.98msec	20.83msec
iteration number	8 10	6 9

Table 5.1: Comparison of device characteristics and testing time for V_t measurement using binary search for two different transistors. The different test times result mainly from the different magnitudes of the target current.

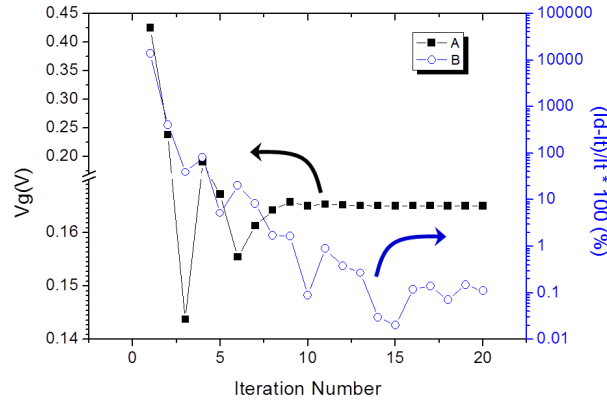


Figure 5.13: Time trace of successive iterations for V_t measurement by binary search, showing gate voltage (left axis) and measured current matching percentage (right axis). I_d is the measured current at the drain node and I_t is the target current for V_t definition by the constant current criteria.

5.3.2 V_t testing time improvement using the interpolation method

According to the analysis in subsection 5.1.2 above, the V_t testing time using the interpolation method can be reduced to 14 and 60 ms for transistors A and B, respectively, because the interpolation method requires only two force-measure iteration cycles, compared to approximately 6–9 for binary search. Fig. 5.14 shows a scatter plot of V_t obtained from the binary search and interpolation methodologies for devices fabricated using an advanced process technology. Evidently the data obtained from these two methods exhibits excellent linearity even though the number of force-measure iterations has been reduced to two by using the interpolation algorithm. In other words, the interpolation algorithm shows no degradation in the accuracy of the V_t measurement, but has a shorter testing time. However, as discussed in the previous section, the interpolation methodology is only suitable for characterization of nearly identical transistor sizes having very similar target current values, and furthermore, these transistors must have no double-hump or similar non-linearities in their I_d - V_g characteristics.

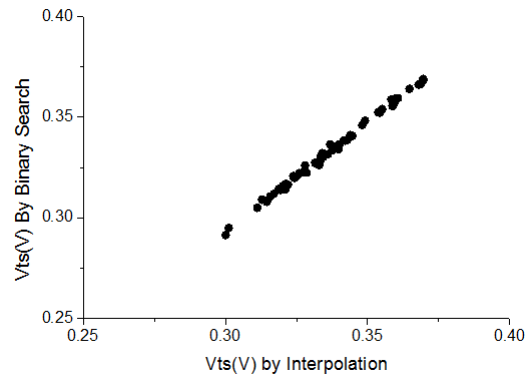


Figure 5.14: Scatter-plot of V_{ts} measurements by the binary search vs the interpolation method.

5.3.3 Simulation of V_t measurement using OP-based SMU

The configuration of Fig. 5.9 was verified by HSPICE simulations in an advanced process technology. In this simulation, a stand-alone DUT (NMOS) is directly connected to a high gain operation amplifier with gain of approximately 100dB. The bias condition of the gate, source and bulk terminals is set as shown in Fig. 5.10 to emulate the test condition. However, in order to trace the voltage modulation at the gate terminal, the drain voltage is swept from 0V to V_{dd} rather than a fixed bias of 0.05V or V_{dd} for V_{tl} or V_{ts} measurement. With the high gain OP, the voltage of the source terminal is clamped at 0V by the virtual short with the OP non-inverting input. Therefore, the absence of body effect in the V_t measurement in this configuration can be verified in simulation if the bulk terminal is also biased at 0V. As plotted in Fig. 5.15, curve A shows that the voltage at the output node of the OP, i.e V_t , is well modulated by a 10 mV change in the drain voltage. Curve B, which represents the voltage at the source terminal, remains at 0V for all values of the drain voltage, indicating that the inverting and non-inverting inputs of the OP amp are a strong virtual short because the gain of the OP amp is sufficiently large. Therefore, by definition, the V_t in the saturation and linear regions can be measured

by forcing $V_d=0.05$ and $V_d=V_{dd}$ respectively, with no inaccuracy introduced by body effect.

The transient simulation for the validation of this array-based test structure also has been checked. Fig. 5.16 shows the simulation waveform of DUT(0,0) - DUT(0,3) in the array-based test structure shown in Fig. 5.12. The voltage-modulated output signal is repeated periodically as the macro scans through the 1024 devices in the array, modulated by the clock period for address switching. These simulations demonstrate the repeatability of successive measurements and the time stability of voltage levels between transitions. This indicates that the OP-based SMU can settle within a clock period of 2~3 ms with 1 pF parasitic capacitance. The V_t measurement demonstrated here is much faster than that of the binary search algorithm.

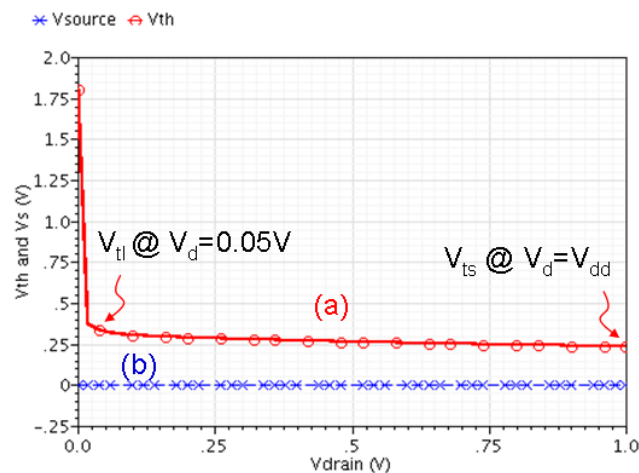


Figure 5.15: Simulation of V_t measurement by an OP-based test structure. Plotted data are (a) OP output voltage, i.e. V_t , and (b) DUT source voltage, which is clamped at 0V due to virtual short to V_{set} .

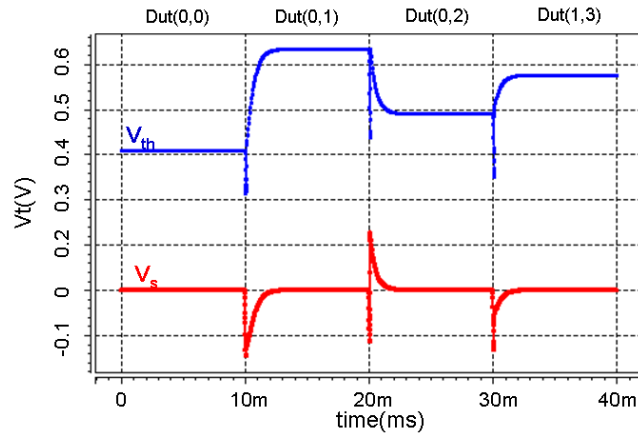


Figure 5.16: Transient simulation of OP-based V_t measurement in an array-based test structure

5.3.4 Stand-alone DUT test result

The proposed OP-based V_t measurement method introduced in Section III-B delivers significant improvement in testing time and measurement accuracy. To demonstrate this, in this experiment, the measurement time and accuracy is evaluated for one of the DUTs fabricated using an advanced process technology. The V_t measurement by the proposed methodology was repeated 1,000 times using an Agilent parametric tester to ensure the statistical significance of the result. As shown by the histogram in Fig. 5.17, the V_t value obtained by the OP-based methodology has a very small measurement error, resulting in a very tight distribution. The standard deviation of the 1,000 V_t measurements is around 0.15 mV, much smaller than the error with 1% matching criteria discussed in subsection 5.3.1. In addition, the test time has been significantly reduced from ~ 10 force-measure iterations to one, i.e from ~ 60 ms to ~ 6 ms for the short channel transistor, which is expected based on the discussion in Section III-A. The major reason for this improvement in testing time is the use of only a single force-measure iteration. Improved measurement accuracy is further ensured by precise setting of the target current and the

absence of impact of errors imperfections and variability of an on-chip OP design, which may occur in an on-chip OP approach such as Ref [88].

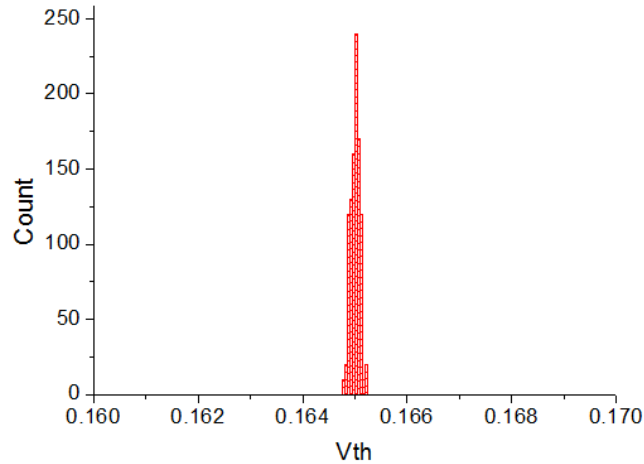


Figure 5.17: V_t distribution obtained by 1000 repeated measurements on the same DUT (shorter channel transistor) by OP-based measurement of the array test structure.

5.3.5 Array-based test structure result

The test speed improvement of OP-based V_t measurement is most significant in array-based test structures. In addition to the test time savings from reducing ~ 10 force-measure iterations to 1, the array-based test structure avoids the time required for the connect and disconnect operations between SMUs and testline IO pads for measurement of successive DUTs which must be performed before the force-measure iterations can begin. Typically, the time required for connect and disconnect operations, which are performed by mechanical switches, is about 1 ms. However, the time required for changing and latching addresses for DUT selection in array-based test structures is less than 1 μ s which is much faster than the connect and disconnect mechanical operations. In practice, for array-based test structure measurement, connection between SMU and pad is performed at the first address and disconnection is performed at the last address because the same SMUs are

used for the force and sense terminals of all DUTs. Moreover, the testing time overhead can be further improved in array-based test structure by elimination of prober index time. In general, the prober index time is typically a few hundred ms if the required prober chuck displacement is less than 1 mm. In this experiment, a test time comparison was performed between (A) non-array-based test structures, and (B) array-based test structure. Both cases include $\sim 1\text{k}$ DUTs with different W/L combinations. However, in case (A), non-array test structures required more a larger number of testlines and thus more layout area. Typically, only 8 DUTs can be placed in one non-array testline due the constraint that DUTs must not have excessive sharing of I/O pads. As a result, 1k DUTs requires $1000/8=125$ testlines in case (A). The V_t of each DUT was measured by the proposed methodology under two different test conditions: 1) non-array DUTs in case A, which require a SMU connect and disconnect for each individual DUT, and 2) an array-based test structure in case (B), which only requires a connect and disconnect for operation for the first and last addresses, respectively. As can be seen in Table II, the time required specifically for the 1k DUT V_t measurements is approximately 6000 ms for both cases (A) and (B). However, case (A) it requires additional overhead of 1000 ms from SMU connect and disconnect operations. Moreover, case A utilizes 125 probe card touch downs instead of the single touch down of case (B). Therefore, case (A) requires 125 prober chuck displacements during measurement. Assuming a prober index time of 200 us, case(A) incurs an additional penalty of approximately 125×200 ms are required for case (A). As a result, the total test time is approximately 32,000 ms and 6,002 ms for the proposed V_t measurement on non-array and array-based test structures, respectively. The test speed is further improved by factor of 5X by taking advantage of a single connect/disconnect and elimination of the prober index time during measurement of the array-based test structure.

	non-array DUT	array-based DUT
# only measurement	~6000 ms	~6000 ms
# connect and disconnect	~1000ms	~2ms
# prober index time	~1000/8*200ms	0ms
# total testing time	~32000sec	~6002ms

Table 5.2: Test time comparison of OP-based Vt measurement between stand-alone DUT and array-based DUT.

Chapter 6

Conclusion

In this thesis, we, first, have presented a novel dummy-fill flow, which applies boolean mask operations to directly generate the mask layers with dummy patterns inserted, and performs dummy generation and mask computation at the foundry simultaneously with dummy fill and post-dummy simulation at the design house. The proposed flow not only improves the efficiency of dummy generation, but also eliminates the delays in tape-out due to dummy insertion and post-dummy simulation, and dramatically reduces the tape-out GDS file size, enabling more rapid first silicon delivery. In comparison with a conventional dummy-fill flow currently widely used throughout the IC industry, the proposed dummy-fill flow can achieve better pattern uniformity with less runtime and smaller GDS-file size. The savings in runtime and GDS-file size will be even more significant in future advanced process technologies as more short-range dummy patterns with smaller pattern dimensions are required. Secondly, we have demonstrated a novel design verification flow, which verifies mask generation algorithms by automated direct comparison with the design netlist utilizing currently available physical verification EDA tools. Minimal effort is required to create both the MVS runset and the virtual mask set GDS for verification. The experiments described above demonstrate that MVS automatically detects common errors. As process technologies continue to scale, and the IC industry is challenged by more complex mask generation and requirements for custom

devices, the proposed design verification flow will become increasingly critical to insuring first silicon success. Next, we successfully developed an array-based test structure and a corresponding novel test methodology for overcoming the IR-drop from parasitic resistance and the leakage current from the control circuitry, which are challenges inherent to any array-based characterization technique. We introduce the technique of hardware IR compensation to address the parasitic IR drop, and the combination of voltage bias elevation and leakage current cancellation to perform efficient, highly accurate current measurements on a large device array. The proposed array-based test structure can fit into the pad frame of a traditional PCM testline and can be placed on a scribe line for production process monitoring. Measurements with the proposed structure have demonstrated accuracy comparable to the PCM testline, but a much larger data volume can be gathered with the same pad frame area. Also, its DUT array size can be further extended for statistical SPICE modeling. A series of experiments were conducted on both mature and newly developed process technologies to validate the effectiveness and the superiority of the overall proposed test structure and methodology. As one example of the application of this technique to perform valuable process characterization, we demonstrate the use of a version of this structure to collect local V_{th} mismatch data for an array of MOSFET pairs. Finally, we successfully developed a testing methodology by using OP-based SMU to speed up the V_{th} measurement. By using the proposed techniques, the V_{th} testing speed can be improved by 5~10 times with better accuracy about 0.15mV. Also, combining with an array-based test structure design, ~ 1k FET the test time of V_t measurements can be further improved by a factor of 5X due to connect, disconnect and prober index time saving. A series of experiments were conducted on both mature and newly developed process technologies to validate the effectiveness and the superiority of the overall proposed test structure as well its application.

Bibliography

- [1] John M. Cohn, David J. Garrod, Rob A. Rutenbar, L. Richard Carley "Analog Device-Level Layout Automation," *Kluwer Academic Publishers Norwell, MA, USA*, 1994.
- [2] Florin Balasa, Sarat C. Maruvada, and Karthik Krishnamoorthy "On the Exploration of the Solution Space in Analog Placement With Symmetry Constraints," *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, pp. 177–191, 2004.
- [3] Huang-Yu Chen, Mei-Fang Chiang, Yao-Wen Chang "Novel Full-Chip Gridless Routing Considering Double-Via Insertion," *IEEE Design Automation Conference*, pp. 755–760, 2006.
- [4] Kuang-Yao Lee, Ting-Chi Wang, Kai-Yuan Chao "Post-Routing Redundant Via Insertion and Line End Extension with Via Density Consideration," *IEEE International Conference on Computer-Aided Design*, pp. 633–640, 2006.
- [5] Olivier Rizzo and Hanno Melzner "Concurrent Wire Spreading, Widening, and Filling," *IEEE Design Automation Conference*, pp. 350–353, 2007.
- [6] Ming-Chao Tsai, Yung-Chia Lin, Ting-Chi Wang "An MILP-Based Wire Spreading Algorithm for PSM-Aware Layout Modification," *IEEE Asia and South Pacific Design Automation Conference*, pp. 364–369, 2008.

- [7] V. Kheterpal, V. Rovner, T. G. Hersan, D. Motiani, Y. Takegawa, A. J. Strojwas, and L. Pileggi "Design methodology for IC manufacturability based on regular logic-bricks" *Annual ACM IEEE Design Automation Conference*, pp. 353–358, 2005.
- [8] Atsush Kurokawa, Toshiki Kanamoto, Akira Kasebe, Yasuaki Inoue, and Hiroo Masuda "Efficient capacitance extraction method for interconnects with dummy fills," *IEEE Custom Integrated Circuit Conference*, pp. 485–488, 2004.
- [9] Andrew B. Kahng, Kambiz Samadi "CMP Fill Synthesis: A Survey of Recent Studies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 3–19, 2008.
- [10] Chih-Ju Hung "Diamond metal-filled patterns achieving low parasitic coupling capacitance," *U.S. Patent 6,998,716*
- [11] Mark M. Nelson "Optimized pattern fill process for improved CMP uniformity and interconnect capacitance," *University/Government/Industry Microelectronics Symposium*, pp. 374–375, 2003.
- [12] Ruiqi Tian, D.F. Wong, Robert Boone "Model-based dummy feature placement for oxide chemical-mechanical polishing manufacturability," *IEEE Design Automation Conference*, pp. 667–670, 2000.
- [13] Yu Chen, Andrew B. Kahng, G. Robins, Alexander Zelikovsky "Practical iterated fill synthesis for CMP uniformity," *IEEE Design Automation Conference*, pp. 671–674, 2000.
- [14] Maharaj Mukherjee, Kanad Chakraborty "A Randomized Greedy Algorithm for the Pattern Fill Problem for DFM Applications," *International Symposium on Quality Electronic Design*, pp. 344–347, 2008.

- [15] Charles Chiang, Jamil Kawa "Design for Manufacturability and Yield for Nano-Scale CMOS," *Springer*, 2007.
- [16] Jianfeng Luo, Qing Su, Charles Chiang, Jamil Kawa "A Layout dependent full-chip copper electroplating topography model," *IEEE International Conference on Computer-Aided Design*, pp. 133–140, 2005.
- [17] Subarna Sinha, Jianfeng Luo, Charles Chiang "Model based layout pattern dependent metal filling algorithm for improved chip surface Uniformity in the copper process," *IEEE Asia and South Pacific Design Automation Conference*, pp. 1–6, 2007.
- [18] R. Li, L. Yu, H. Xin, Y. Dong, K. Tao, and C. Wang "A comprehensive study of reducing the STI mechanical stress effect on channel-width-dependent Idsat" *Semiconductor Science and Technology*, Vol.22, pp. 1292–1297, 2007.
- [19] Oliver Pohland, Julie Spieker, Chih-Ta Huang, Srikanth Govindaswamy, and Artur Balasinski "New type of dummy layout pattern to control ILD etch rate" *Proceedings of the SPIE*, , Vol. 6798, pp.679804, 2007.
- [20] Nakao, Shuji; Tsujita, Kouichirou; Arimoto, Ichirou; Wakamiya, Wataru "0.32-um pitch random line pattern formation by dense dummy pattern and double exposure in KrF wavelength" *Proceedings of the SPIE*, , Vol. 4000, pp. 1123–1133, 2000.
- [21] Brent A. Anderson, Howard S. Landis, and Edward J. Nowak, "Variable overlap of dummy shapes for improved rapid thermal anneal uniformity" *US patent 7,537,941 B2*
- [22] Laurent Remy, Philippe Coll, Fabrice Picot "Metal Filling Impact on Standard Cells : Definition of the Metal Fill Corner Concept," *Proceedings of the 21st*

annual symposium on Integrated circuits and system design, September 1–4, 2008.

- [23] Oliver Pohland, Julie Specker, Chih-Ta Huang, Srikanth Govindaswamy, and Arthur Balasinski "New Type of Dummy Layout Pattern to Control ILD etch Rate," *Proc. SPIE Vol. 6798, 679804*, 2007.
- [24] Y. Uematsu, "Integrated circuit device having dummy pattern effective against micro loading effect" *US Patent 5,598,010*
- [25] James A. Wilmore "Efficient boolean operations on IC masks," *IEEE Design Automation Conference*, pp. 571–579, 1981.
- [26] Ulrich Lauther "An $O(N \log N)$ algorithm for boolean mask operations," *IEEE Design Automation Conference*, pp. 555–562, 1981.
- [27] Young-Mi Kim, Sang-Uk Lee, Jea-Hyun Kang, Jea-Hee Kim, and Kee-Ho Kim "Application of modified jog-fill DRC rule on LFD OPC flow" *Photomask technology. Conference*, Vol. 6730(3), pp.67303W.1–67303W, 2007.
- [28] Todd J. Wagner, "A HIERARCHICAL APPROACH FOR LAYOUT VERSUS CIRCUIT CONSISTENCY CHECK" *17th conference on Design*, pp.270-276, 1980
- [29] Todd J. Wagner, "HIERARCHICAL LAYOUT VERIFICATION" *Design Automation Conference*, pp.848-849, 1984
- [30] Mohy, Ahmed ; Makarem, Mohamed Abul, "A robust and automated methodology for LVS quality assurance" *Design and Test Workshop (IDT)*, 2009 4th International

- [31] Watts, J. Ke-Wei Su Basel, M. "Netlisting and Modeling Well-Proximity Effects" *Electron Devices, IEEE Transactions on devices*, 2006 Volume: 53, Issue: 9 2179-2186
- [32] Graphics, M. (n.d.). "Reduce Cycle Time with Revolutionary New Capabilities from Calibre nmDRC" *Mentor*, October 20, 2009
- [33] Michael Orshansky, Sani R. Nassif, and Duane Boning, Design for Manufacturability and Statistical Design, A Constructive Approach, *ISBN 978-0-387-30928-6*
- [34] James A. Wilmore "Efficient boolean operations on IC masks," *IEEE Design Automation Conference*, pp.571-579, 1981.
- [35] Wlrich Lauther "An $O(N \log N)$ algorithm for boolean mask operations," *IEEE Design Automation Conference*, pp.555-562, 1981.
- [36] Huang-Yu Chen, Mei-Fang Chiang, Yao-Wen Chang "Novel Full-Chip Gridless Routing Considering Double-Via Insertion," *IEEE Design Automation Conference*, pp.755-760, 2006.
- [37] Kuang-Yao Lee, Ting-Chi Wang, Kai-Yuan Chao "Post-Routing Redundant Via Insertion and Line End Extension with Via Density Consideration," *IEEE International Conference on Computer-Aided Design*, pp.633-640, 2006.
- [38] Olivier Rizzo and Hanno Melzner "Concurrent Wire Spreading, Widening, and Filling," *IEEE Design Automation Conference*, pp.350-353, 2007.
- [39] Ming-Chao Tsai, Yung-Chia Lin, Ting-Chi Wang "An MILP-Based Wire Spreading Algorithm for PSM-Aware Layout Modification," *IEEE Asia and South Pacific Design Automation Conference*, pp.364-369, 2008.

- [40] Ghani, T. et al; "A 90nm high volume manufacturing logic technology featuring novel 45nm gate length strained silicon CMOS transistors," *IElectron Devices Meeting, 2003.*, pp.11.6.1-11.6.3, 2003.
- [41] Neil Weste, David Harris; "CMOS VLSI Design: A Circuits and Systems Perspective (3rd Edition)," *ISBN: ISBN-13: 978-0321149015*,
- [42] Diaz, C.H. et al; "32nm gate-first high-k/metal-gate technology for high performance low power applications," *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, pp.11-4, 2008.
- [43] X. Chen et al; "A cost-effective 32nm High-K metal-gate CMOS technology for low-power applications with single-metal/gate-first process," *VLSI Tech. Symp.*, pp.88-89, 2008.
- [44] Chris Spence et al; "Mask data volume: historical perspective and future requirements," *22nd European Mask and Lithography Conference* , Proc. SPIE, Vol. 6281.
- [45] Graphics, M. (n.d.). "Appendix C: GDS II Format. Retrieved October 20, 2009" *Computer Aids for VLSI Design*, 1994
- [46] Graphics, M. (n.d.) "Reduce Cycle Time with Revolutionary New Capabilities from Calibre nmDRC," http://www.mentor.com/products/ic_nanometer_design/verification-signoff/physical-verification/calibre-nmdrc/, October 20, 2009
- [47] "Technology-Leading Physical Verification Solution for 45nm and Above " <http://www.synopsys.com/tools/implementation/physicalverification/pages/hercules.aspx>, 2009

- [48] K. Bernstein, et al., "High-performance CMOS variability in the 65-nm regime and beyond", *IBM Journal of Research and Development*, vol.50, no.4.5, pp.433-449, July 2006.
- [49] Michihiro Kanno, et al., "Empirical Characteristics and Extraction of Overall Variations for 65-nm MOSFETs and Beyond", *Empirical Characteristics and Extraction of Overall Variations for 65-nm MOSFETs and Beyond*, VLSI Technology, 2007 IEEE Symposium on, vol., no., pp.88-89, 12-14 June 2007.
- [50] Michael Orshansky, et al. "Design for Manufacturability and Statistical Design, A Constructive Approach", ISBN 978-0-387-30928-6.
- [51] B. Stine, et al. "Analysis and Decomposition of Spatial Variation in Integrated Circuit Processes and Devices", *Semiconductor Manufacturing, IEEE Transactions on*, vol.10, no.1, pp.24-41, Feb 1997.
- [52] M. Orshansky, L. Milor and C. Hu, "Characterization of Spatial Intrafield Gate CD Variability, Its Impact on Circuit Performance, and Spatial Mask-Level Correction", *Semiconductor Manufacturing, IEEE Transactions on*, vol.17, no.1, pp. 2- 11, Feb. 2004.
- [53] Masaharu Yamamoto, et al. "Development of a Large-Scale TEG for Evaluation and Analysis of Yield and Variation", *Semiconductor Manufacturing, IEEE Transactions on*, vol.17, no.2, pp. 111- 122, May 2004.,
- [54] Nigel Drego, Anantha Chandrakasan, and Duane Boning, "A Test-Structure to Efficiently Study Threshold-Voltage Variation in Large MOSFET Arrays", *Quality Electronic Design, 2007. ISQED '07. 8th International Symposium on*, vol., no., pp.281-286, 26-28 March 2007.,

- [55] Sang-Hoon Lee, *Dong-Yun Lee, Tae-Jin Kwon, Joo-Hee, Young-Kwan Park, "An Efficient Statistical Model using Electrical Tests for GHz CMOS Devices", *Statistical Metrology, 2000 5th International Workshop on*, vol., no., pp.72-75, 2000.
- [56] Keiji Nagase, Shin-ichi Ohkawa, Masakazu Aoki, and Hiroo Masuda, "Variation Status in 100nm CMOS Process and Below", *Microelectronic Test Structures, 2004. Proceedings. ICMTS '04. The International Conference on*, vol., no., pp. 257- 261, 22-25 March 2004.
- [57] H. Tsuno, et al. "Advanced Analysis and Modeling of MOSFET Characteristic Fluctuation Caused by Layout Variation", *VLSI Technology, 2007 IEEE Symposium on*, vol., no., pp.204-205, 12-14 June 2007.
- [58] Kanak Agarwal, et al. "A Test Structure for Characterizing Local Device Mismatches", *2006 SVLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, vol., no., pp.67-68, 0-0 0.
- [59] Y. Z. Xu, C. S. Chen, J. I. Watt, "Investigation of 65 nm CMOS transistor local variation using a FET array", *Solid-State Electronics*, pp. 1244-1248, 2008,
- [60] Chris Jakubiec, et al. "An Integrated Test Chip for the complete Characterization and Monitoring of a 0.25um CMOS Technology that fits into scribe line structures 150um by 5,000um" , *Microelectronic Test Structures, 2003. International Conference on*, vol., no., pp. 3- 63, 17-20 March 2003.,
- [61] Naoki Izumi, Hiroji Ozaki, Yoshikazu Nakagawa, Naoki Kasai, and Tsunetoshi Arikado, "Evaluation of Transistor Property Variations Within Chips on 300-mm Wafers Using a New MOSFET Array Test Structure", *Semiconductor Manufacturing, IEEE Transactions on*, vol.17, no.3, pp. 248- 254, Aug. 2004.,

- [62] K. Doong, et al., "Field-Configurable Test Structure Array (FC-TSA): Enabling Design for Monitor, Model, and Manufacturability", *Semiconductor Manufacturing, IEEE Transactions on* , vol.21, no.2, pp.169-179, May 2008.,
- [63] R.W. Gregor, et al., "On the relationship between topography and transistor matching in an analog CMOS technology", *IElectron Devices, IEEE Transactions on* , vol.39, no.2, pp.275-282, Feb 1992.,
- [64] C Abel, C. Michael, M. Ismail, C. S. Teng, R. Lahri, "Characterization of transistor mismatch for statistical CAD of submicron CMOS analog circuits", *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on* , vol., no., pp.1401-1404, 3-6 May 1993.,
- [65] J. Bastos, M. Steyaert, B. Graindourze, W. Sansen, "Matching of MOS transistors with different layout styles", *ICMTS 1996. Proceedings. 1996 IEEE International Conference on* , vol., no., pp.17-18, 25-28 Mar 1996.,
- [66] Pelgrom MJM, Duinmaijer ACJ, Welbers APG, "Matching Properties of MOS Transistors", *Matching properties of MOS transistors," Solid-State Circuits, IEEE Journal of* , vol.24, no.5, pp. 1433- 1439, Oct 1989.
- [67] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, N. J. Rohrer, High-performance CMOS variability in the 65-nm regime and beyond, *IBM J. RES. DEV. VOL. 50 NO. 4/5 JULY/SEPTEMBER 2006*
- [68] Michihiro Kanno, Akira Shibuya, Masao Matsumura, Kazuhiro Tamura, Hitoshi Tsuno, Shigetaka Mori, Yuzo Fukuzaki, Tetsuo Gocho, Hisahiro Ansai and

- Naoki Nagashima, Empirical Characteristics and Extraction of Overall Variations for 65-nm MOSFETs and Beyond, *2007 Symposium on VLSI Technology Digest of Technical Papers*
- [69] Michael Orshansky, Sani R. Nassif, and Duane Boning, Design for Manufacturability and Statistical Design, A Constructive Approach, *ISBN 978-0-387-30928-6*
- [70] B. Stine, D. Boning and J. Chung, Analysis and Decomposition of Spatial Variation in Integrated Circuit Processes and Devices, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 10, No 1., 1997.
- [71] M. Orshansky, L. Milor and C. Hu, Characterization of Spatial Intrafield Gate CD Variability, Its Impact on Circuit Performance, and Spatial Mask-Level Correction, *IEEE Trans. On Semiconductor Manufacturing*, Vol. 12, No. 1, 2004.
- [72] Masaharu Yamamoto, Development of a Large-Scale TEG for Evaluation and Analysis of Yield and Variation, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL. 17, NO. 2, MAY 2004
- [73] Nigel Drego, Anantha Chandrakasan, and Duane Boning, A Test-Structure to Efficiently Study Threshold-Voltage Variation in Large MOSFET Arrays, *Proceedings of the 8th International Symposium on Quality Electronic Design (ISQED'07)*,
- [74] Sang-Hoon Lee, *Dong-Yun Lee, Tae-Jin Kwon, Joo-Hee, Young-Kwan Park, An Efficient Statistical Model using Electrical Tests for GHz CMOS Devices, *2000 5th INTERNATIONAL WORKSHOP ON STATISTICAL METROLOGY 2000*;72-75

- [75] Keiji Nagase, Shin-ichi Ohkawa, Masakazu Aoki, and Hiroo Masuda, Variation Status in 100nm CMOS Process and Below, *Proc. IEEE 2004 Int. Conference on Microelectronic Test Structures, Val 17. March 2004; 257-261*
- Ghani, T. (Intel) et al., A 90nm high volume manufacturing logic technology featuring novel 45nm gate length strained silicon CMOS transistors; 2003 IEEE International Electron Devices Meeting, IEDM '03 Technical Digest. 8-10, Dec. 2003 Page(s):11.6.1 - 11.6.3.
- [76] Ghani, T et al., A 90nm high volume manufacturing logic technology featuring novel 45nm gate length strained silicon CMOS transistors, *2003 IEEE International Electron Devices Meeting, IEDM '03 Technical Digest. 8-10, Dec. 2003 Page(s):11.6.1 - 11.6.3.*
- [77] Chien-Hao Chen et al., Stress memorization technique (SMT) by selectively strained-nitride capping for sub-65nm high-performance strained-Si device application, *2004. Digest of Technical Papers.15-17 June 2004 Page(s):56 V57*
- [78] H. Tsuno, K. Anzai, M. Matsumura, S. Minami, A. Honjo, H. Koike, Y. Hiura, A. Takeo, W. Fu, Y. Fukuzaki, M. Kanno, H. Ansai and N. Nagashima, Advanced Analysis and Modeling of MOSFET Characteristic Fluctuation Caused by Layout Variation, *2007 Symposium on VLSI Technology Digest of Technical Papers 204-205*
- [79] Kanak Agarwal, Frank Liu, Chandler McDowell, Sani Nassif, Kevin Nowka, Meghann Palmer, Dhruva Acharyya, Jim Plusquellic, A Test Structure for Characterizing Local Device Mismatches, *2006 Symposium on VLSI Circuits Digest of Technical Papers*

- [80] Y. Z. Xu, C. S. Chen, J. I. Watt, Investigation of 65 nm CMOS transistor local variation using a FET array, *Soild-State Electrinics*, 52 (2008) 1244-1248
- [81] Chris Jakubiec, An Integrated Test Chip for the complete Characteriztion and Monitoring of a 0.25um CMOS Technology that fits into scribe line structures 150um by 5,000um, *ICMTS*, 2003,03 59-63
- [82] Naoki Izumi, Hiroji Ozaki, Yoshikazu Nakagawa, Naoki Kasai, and Tsunetoshi Arikado, Evaluation of Transistor Property Variations Within Chips on 300-mm Wafers Using a New MOSFET Array Test Structure, *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, VOL. 17, NO. 3, AUGUST 2004
- [83] K. Doong, et al., Field-Configurable Test Structure Array (FC-TSA): Enabling Design for Monitor, Model, and Manufacturability, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 21, N o. 2, May 2008
- [84] Tseng-Chin Luo, Mango C.-T. Chao, et al; A novel array-based test methodology for local process variation monitoring , *ITC 2009*
- [85] J. A. Croon, H. P. Tuinhout, R. Difrenza, J. Knol, A. J. Moonen, S. Decoutere, H. E. Maes, and W. Sansen, A comparison of extraction techniques for threshold voltage mismatch , *Proc. 2002 Int. Conf. Microelectronic Test Structures, ICMTS-2002*, pp. 235-240.
- [86] H. Lee, S. Oh, and G. Fuller, A simple and accurate method to measure the threshold voltage of an enhancement-mode MOSFET , *IEEE Trans. Electron Devices*, vol. 29, p. 346, 1982
- [87] S. Wolf Silicon Processing for the VLSI Era Volume 3 - The Submicron MOS-FET, *ISBN 0-9616721-4-5*

- [88] Brian L. Ji, et al; Operational Amplifier Based Test Structure for Quantifying Transistor Threshold Voltage Variation , *IEEE TRANSACTIONS ON SEMI-CONDUCTOR MANUFACTURING*, VOL. 22, NO. 1, FEBRUARY 2009
- [89] YUAN TAUR, TAK H,NING Fundamentals of Modern VLSI DEVICES, *ISBN 0-521-55056-4*



Vita



Tseng-Chin Luo received the M.S degree in Material Science and Engineering from National Chiao-Tung University in 1994, and is currently pursuing the Ph.D degree in Electrical Engineering at the same university. He joined Winbond Electronics, Hsinchu, Taiwan, in 1996, and then transferred to Worldwide Semiconductor Manufacturing Corporation (WSMC), Hsinchu, Taiwan, in 1997. His major focus was parametric testing and process integration during this period. Since 1998, he has been developing processes for .18um and .13um technology in the Logic Technology Research and Development Division at Taiwan Semiconductor Manufacturing Corp (TSMC). Recently, he has focused on developing test structures and fast test methodology for process characterization, design manufacturing and yield optimization, establishing infrastructure to improve design effectiveness and analysis quality. He is now Project Manager of Fast Parametric Test Solutions at TSMC.

Publication List

- Journal Paper:

- Tseng-Chin Luo and Chao, M.C.-T, “A novel array-based test methodology for local process variation monitoring,” *IEEE Transactions on Semiconductor Manufacturing* , vol.PP, no.99.
- Tseng-Chin Luo and Chao, M.C.-T, “A Novel Design Flow for Dummy Fill Using Boolean Mask Operations,” *IEEE Transactions on Semiconductor Manufacturing(Accepted)*

- Conference Paper:

- Tseng-Chin Luo and Chao, M.C.-T, “A novel array-based test methodology for local process variation monitoring,” *Test Conference, 2009. ITC 2009. International*, vol., no., pp.1-9, 1-6 Nov. 2009.
- Tseng-Chin Luo and Chao, M.C.-T, “Mask versus Schematic – An Enhanced Design-Verification Flow for First Silicon Success,” *Test Conference (ITC), 2010 IEEE International* , vol., no., pp.1-9, 2-4 Nov. 2010.