Improved effective channel length extraction method for 0.09-0.13 mm CMOS

Eng, Chee Wee

2008

Eng, C. W. (2008). Improved effective channel length extraction method for 0.09-0.13 mm CMOS. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4118>

<https://doi.org/10.32657/10356/4118>

Nanyang Technological University

Downloaded on 26 Apr 2023 15:31:37 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**AN IMPROVED EFFECTIVE CHANNEL LENGTH
EXTRACTION METHOD for 0.09-0.13 μm CMOS**

ENG CHEE WEE

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2008

**An Improved Effective Channel Length
Extraction Method
for 0.09-0.13 μm CMOS**

Eng Chee Wee

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

Acknowledgments

I wish to express my sincere gratitude to my advisor Associate Professor Lau Wai Shing for his dedicated guidance and assistance given to me throughout this research project in Nanyang Technological University.

I would also like to thank Dr. Lap Chan from Chartered Semiconductor Manufacturing Ltd. for his guidance, suggestions and support throughout my research at NTU and Chartered Semiconductor and Mr. David Vigar for his continuous support and faith in me. I would like to thank Dr. Alex See and Mr. Tee Kheng Chok for their constructive suggestions and helpful discussion. I am grateful to Mdm Jiang Yao Yao, Mdm Vanissa Lim and Dr. Alastair Trigg from the Institute of Microelectronics for their excellent technical support regarding Scanning Capacitance Microscopy.

Some of my sweetest memories at Chartered Semiconductor are of my fellowship with those graduate students in the Special Project group. I would like to thank Chua Gek Soon, Jason Tan, Serene Chan, Neo Chin Chuan, Teo Tai Yong, Clare, Tan Sia Kim and Pang Chong Hou. They have brought insight, intelligence, entertainment, and friendship, which have made my life at Chartered Semiconductor a memorable one.

I would like to express my deeply gratitude to my family, especially my parents (Eng Siow Hwee and Poh Ah Hua) for their love, care, understanding, support and encouragement during all these years. I would also like to thank Nanyang Technological University for research scholarship and Chartered Semiconductor Manufacturing Limited for additional financial assistance.

Table of contents

Acknowledgments	i
Table of Contents	ii
Summary	v
List of Figures	vii
List of Tables	ix

1. Introduction

1.1	MOSFET scaling and device characterization challenges	1
1.2	Issues and limitation of current method of L_{eff} extraction with channel length scaling	4
1.3	Motivation and objectives	8
1.4	Organization and contribution of the thesis	8

2. Review of L_{eff} Extraction Methods for MOSFET

2.1	Introduction: Semiconductor device characterization	10
2.2	Various definitions of channel length	11
2.3	Electrical characterization- Extraction of L_{eff}	13
2.3.1	Channel resistance based method	14
2.3.1.1	Channel resistance method by Chern et al. in 1980 ...	17
2.3.1.2	Shift-and-Ratio method proposed by Taur et al.	18
2.3.1.3	Limitation of current <i>Channel Resistance</i> method of L_{eff} Extraction	21
2.3.2	Capacitance based method	23
2.3.3	DIBL method	26
2.3.4	Direct tunneling method	27
2.4	Physical characterization	30

2.4.1	Scanning capacitance microscopy	31
2.4.2	Scanning electron microscopy	36
2.4.3	Transmission electron microscopy	39
2.5	Electron mobility of MOSFET's with ultrathin gate oxide	43
2.6	Chapter summary	48
3.	Theory of a New L_{eff} Extraction Method	
3.1	Introduction	50
3.2	Theory of our new approach	57
3.3	Remote charge scattering at low gate voltage	64
3.4	Device fabrication and experiment setup	66
3.5	New methodology	68
3.5.1	Methodology I: A Modified Shift-and-Ratio Method	68
3.5.2	Methodology II: A Modified Shift-and-Ratio Method for extraction of relative change in L_{eff}	72
3.5.3	Methodology III: A Subthreshold Regime-based Method ...	73
3.6	Chapter summary	80
4.	Result and Discussion on Modified Shift-and-Ratio Method	
4.1	Introduction.....	83
4.2	Result and discussion I	84
4.2.1	Carrier mobility estimation	94
4.2.2	Consistency check using SCM measurement	97
4.2.3	L_{eff} extraction at low bias gate voltage range	105
4.2.4	L_{eff} correlation with I_{off}	107
4.3	Result and discussion II	116

4.4	Chapter summary	126
5.	Result and Discussion on Subthreshold Regime-Based Method	
5.1	Result and discussion III	129
5.2	Chapter summary	137
6.	Conclusions and Recommendation	
6.1	Conclusions	139
6.1.1	A review of goals and achievement	139
6.1.2	List of publications	141
6.2	Recommendations for future work	143
6.2.1	Application and improvement of our L_{eff} extraction method	143
6.2.2	Theoretically calculation of remote Coulomb scattering	144
6.2.3	Extension of our L_{eff} extraction method to latest technology	144
	Bibliography	
	Appendix A	
	Appendix B	

Summary

Over the years, substantial research effort has been focused on developing a simple yet effective method to extract the effective channel length, L_{eff} . Accurate extraction of L_{eff} in MOS transistors is extremely important for both process monitoring/control and device/circuit design as devices are scaled down into the sub-100 nm regime.

In early works of L_{eff} extraction, the MOS transistors under test had a relatively simple source-drain (S/D) structure with an infinitely abrupt (laterally) S/D junction. As MOS transistors scaled, one of the key difficulties is the control of short channel effect (SCE). Over the years, to overcome SCE, the MOS transistor structure had migrated from relatively early MOSFET's with a simple S/D structure to more advanced MOSFET's with shallow, lightly doped S/D extensions, formed by implantation of arsenic for n-channel and boron for p-channel MOS transistor. In addition, a halo or pocket implant is used as a punchthrough stopper and also for suppressing SCE. This halo or pocket implant is particularly important for very small gate length transistors. However, due to this implant, the short channel transistor has a smaller effective mobility compared to the long channel reference transistor. L_{eff} extraction approaches based on the I-V characteristics at low drain bias (or linear region) tend to overestimate the value of L_{eff} . This is because for conventional *channel resistance* L_{eff} extraction method, the effective mobility was assumed to be independent of channel length. Other approaches may involve algorithms which may be too complicated and difficult to implement in automatic wafer electrical testing systems for a large volume of wafer testing.

It has been the purpose of this work to explore ways to overcome the limitation of existing L_{eff} extraction methods thus develop method that can be easily integrated into automatic wafer electrical test systems for routine process monitoring. This work involves the detailed study and understanding of physical limits and shortcoming of existing electrical parameter extraction methodology. With the understanding of the limitation and shortcoming of existing electrical parameter extraction methodology, this work further propose to make the L_{eff} extraction near the threshold voltage regime. In the gate voltage region near the threshold voltage, the carrier mobility is approximately equal for both the short and long channel transistors due to *remote charge scattering (RCS)*, thus, overcomes the limit of existing L_{eff} extraction method. In addition, through the discovery of this new device physics for state-of-the-art metal-oxide-semiconductor field-effect-transistors (MOSFETs) with ultra-thin gate oxide (≤ 2 nm), an original and novel method of L_{eff} extraction based on subthreshold regime has been developed. This Subthreshold Regime-based method was successfully applied on advanced 0.13 μm , 0.11 μm , 0.1 μm and 90nm MOS transistors. This demonstrates the usefulness and capability of the new method in extracting L_{eff} successfully on advanced MOS transistors with halo/pocket implant

List of figures

Figure		Page
1.1	Exponential growth of the number of components per IC chip versus year for (a) Si-based MOS, (b) Si-based bipolar, (c) III-V semiconductor based MESFET, and (d) III-V semiconductor based MODFET [2].	2
1.2	Exponential decrease of (a) minimum feature length, (b) junction depth, and (c) gate oxide thickness of MOSFET [2].	3
1.3	The evolution of MOSFET structures as shown in schematic cross-sections. (a) simple S/D MOSFET, (b) LDD MOSFET with relatively uniform channel doping, (c) MOSFET with punchthrough stopper used to suppress punchthrough and also <i>SCE</i> , (d) MOSFET with punchthrough stopper and also halo/pocket implant used to further suppress punchthrough and also <i>SCE</i> .	5
1.4	Schematic diagrams showing MOS transistors with halo implants (a) when the channel length is large and (b) when the channel length is small. It can be easily seen that for short MOS transistors, the halo implant from the source side overlaps with the halo implant from the drain side.	7
2.1	Schematic diagram showing the relationship among the various definitions of channel length [24].	12
2.2	Schematic diagram of MOS transistor with S/D series resistance [24].	16
2.3	L_{eff} extraction method modified from the original method proposed by Chern et al. [26]	17
2.4	(a) $S(V_{gs})$ curves for long-channel and short-channel devices. The device width is 10 μm . (b) Average ratio (open circle) and variance (solid dot) versus gate voltage shift [29].	20
2.5	Total resistance $R_{TOT} = V_{DS}/I_{DS}$ of scaled devices biased at the same gate overdrive $V_{GT} = (V_{GS} - V_T)$: Dashed lines represent the linear regression of R_{TOT} (at the three different values of V_{GT}), calculated considering only the data measured on MOSFET's with L_{mask} ranging from 0.35 to 0.9 μm [59].	23
2.6	Fig. 2.6. Simulated $C_{dg} + C_{sg}$ versus V_g for r-MOSFETs with different gate lengths. This plot illustrates the separation between intrinsic and extrinsic regions. In particular the bias dependent extrinsic capacitance	25

	is underlined. Inset: MOSFET structure generated with the 2D process simulator [62].	
2.7	Example of $DV_{t_{dibl}}$ plotted versus $L_{poly(cap)}$ for NMOSFET devices. The trend line shows the exponential dependence of $DV_{t_{dibl}}$ on $L_{poly(cap)}$ [49].	27
2.8	Schematic view of direct tunneling current (DTC) of a MOSFET.	28
2.9	Measured and modeled direct tunneling current at $V_{ds}=0.05V$ at different channel length. [66]	29
2.10	Optical lever of AFM. [71].	31
2.11	SCM detection system schematic.	33
2.12	Capacitance vs. voltage curve for n-type semiconductor material.	34
2.13	Change from accumulation to depletion during SCM due to alternating electric field.	35
2.14	Schematic of a scanning electron microscope	38
2.15	Schematic of diffraction pattern, projecting the diffraction pattern on the viewing screen.	41
2.16	Electron mobility in inversion layer at 300K and 77K versus effective electric field, $E_{eff} = q(N_{dpl} + N_s)/e_{si}$ with N_s of $1/2$. [79].	44
2.17	Schematic diagram showing the E_{eff} or N_s dependence of mobility in inversion layer by three dominant scattering mechanisms [79].	45
2.18	Inversion-layer electron mobility as a function of E_{eff} with T_{ox} of 3.5–1.5 nm. The solid line denotes the universal mobility curve [88].	47
3.1	Schematic drawing of the cross section view of the transistor where the concentrations of the ions are extracted.	51
3.2	One Dimensional (1-D) Channel doping profile of (a) long channel (10 μm) and (b) short channel (0.13 μm) NMOS transistors.	52
3.3	One-dimensional (1-D) channel doping profile of (a) long channel (10 μm) and (b) short channel (0.13 μm) PMOS transistors.	54
3.4	L_{eff} extracted using <i>channel resistance</i> method by Chern <i>et al.</i> [26] for NMOS transistors with channel width of 20 μm of 0.13 μm technology.	55
3.5	L_{eff} extracted using <i>channel resistance</i> method [26] for NMOS	56

transistors with channel width of 10 μm of 0.13 μm technology.

3.6	Schematic diagram of the channel region in a n-channel MOSFET with relatively thick gate oxide (a) Low channel doping concentration; (b) High channel doping concentration.	58
3.7	Schematic diagram of the channel region in a n-channel MOSFET with ultra-thin gate oxide (a) Low channel doping concentration; (b) High channel doping concentration.	58
3.8	Mobility versus gate oxide thickness plot modified from Fig.10. of [84].	60
3.9	Energy band diagram of a cross-section of a n-channel MOSFET. (a) at $V_{gs} = 0$; (b) V_{gs} slightly greater than V_{th} ; (c) V_{gs} strongly greater than V_{th} .	62
3.10	Energy band diagram of a cross-section of a p-channel MOSFET. (a) at $V_{gs} = 0$; (b) V_{gs} slightly greater than V_{th} ; (c) V_{gs} strongly greater than V_{th} .	63
3.11	(a) RCS and CS components of mobility close to threshold, (b) Mobility of short and long MOS transistors close to threshold.	66
3.12	Average ratio (open dot) and variance (solid dot) versus gate voltage shift. Variance spread over some values, difficult to determine the correct minimum point for constant ratio.	70
3.13	Example of sub-threshold swing (S) plotted versus L_{drawn} for NMOS transistor. It shows an exponential dependence of S on L_{drawn} .	74
3.14	Drift and diffusion components of current in an I_{ds} - V_g plot. Their sum is a total current represented by the solid curve [101].	75
3.15	Nonlinear least squares fitting of the subthreshold current of a long channel transistor. The value generated for α is equal to 3.27×10^{-7} and V_{th} is 0.29 V.	79
3.16	Nonlinear least squares fitting of the subthreshold current of a short channel transistor. The value generated for α is equal to 8×10^{-5} and V_{th} is 0.33 V	80
4.1	Fig. 4.1: Extracted L_{eff} with $V_d = 0.05$ V for various V_{gs} ranges of NMOS transistor for 0.13 μm CMOS technology	84
4.2	Extracted L_{eff} with $V_d = 0.1$ V for various V_{gs} ranges of NMOS transistor for 0.13 μm CMOS technology.	85

4.3	Extracted L_{eff} with $V_d = 0.05$ V for various V_{gs} ranges of PMOS transistor for 0.13 μ m CMOS technology.	85
4.4	Extracted L_{eff} with $V_d = 0.1$ V for various V_{gs} ranges of PMOS transistor for 0.13 μ m CMOS technology.	86
4.5a	Extracted channel length, L_{eff} , and delta L , DL , of NMOS transistors using modified method for 0.13 μ m technology. (solid dot) represents the L_{eff} and (open dot) represents the ΔL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.018 μ m.	87
4.5b	Extracted channel length, L_{eff} , and delta L , DL , of NMOS transistors using modified method for 0.1 μ m technology. (solid dot) represents the L_{eff} and (open dot) represents the ΔL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.008 μ m.	88
4.6a	Extracted channel length, L_{eff} , and delta L , ΔL , of PMOS transistors using modified method for 0.13 μ m technology. In the figure, (solid dot) represents the L_{eff} and (open dot) represents the ΔL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.033 μ m.	89
4.6b	Extracted channel length, L_{eff} , and delta L , ΔL , of PMOS transistors using modified method for 0.1 μ m technology. In the figure, (solid dot) represents the L_{eff} and (open dot) represents the ΔL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.015 μ m.	70
4.7	Fig. 4.7: Mobility versus $V_{gs} - V_{th}$ obtained by a rough estimation method for (a) NMOS and (b) PMOS of various channel lengths	96
4.8	Cross-sectional scanning capacitance microscopy image of a deep-submicron PMOS transistor.	99
4.9	A p-channel MOSFET: Topography image acquired during SCM scanning (left) and TEM micrograph (right). Difference between the distance AB obtained in topography image and TEM micrograph was used to correct the error of SCM measurement.	99
4.10	Box plot of L_{eff} extracted using V_{gs} range from 0.2-0.5 V for both NMOS (shaded) and PMOS (unshaded) transistors for (a) 0.13 μ m CMOS technology and (b) 0.1 μ m CMOS technology. The magnitude of V_{ds} was set at 0.05 V.	101

4.11	Plot of L_{eff} vs. L_{SCM} of the NMOS and PMOS transistors for both 0.13 μm and 0.1 μm CMOS technology.	103
4.12	I-V characteristics of two NMOS transistors with the same L_{drawn} but different L_{eff} measured by our modified shift-and-ratio method with a V_{gs} range from 0.2 to 0.5 V, showing that a higher off current and a higher on current correspond to a smaller L_{eff} measured.	104
4.13	L_{eff} extracted for different V_{gs} range for 0.13 μm CMOS technology NMOS transistors. V_{ds} was biased at 0.05 V.	105
4.14	DL extracted for different V_{gs} range for 0.13 μm CMOS technology PMOS transistors. V_{ds} was biased at 0.05 V.	106
4.15	(a) Variation of I_{eff} in I_{on} -log I_{off} relationship (b) Variation of DIBL in I_{on} -log I_{off} relationship.	108
4.16	I_{on} versus DIBL for both NMOS and PMOS transistors	108
4.17	Logarithm of I_{off} versus DIBL for both NMOS and PMOS transistors	109
4.18	Plot of L_{eff} vs. I_{off} for NMOS transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method (0.1-0.5 V V_{gs} range) and also the capacitance method. (Note: CC is correlation coefficient.)	111
4.19	Plot of L_{eff} vs. I_{off} for NMOS transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method at 3 different V_{gs} ranges and also by the S&R method for one V_{gs} range. (Note: CC is correlation coefficient.)	112
4.20	Plot of L_{eff} vs. I_{off} for PMOS transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method (0.1-0.5 V V_{gs} range) and the capacitance method. (Note: CC is correlation coefficient.)	113
4.21	Plot of L_{eff} vs. I_{off} for PMOS transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method for 3 different V_{gs} ranges and also by the S&R method for one V_{gs} range. (Note: CC is correlation coefficient.)	114
4.22	Plot of I_{off} vs. I_{eff} for NMOS and PMOS transistors fabricated by 90 nm CMOS technology with L_{gate} of 60nm. L_{eff} was extracted by MS&R method (0.1-0.3 V V_{gs} range), (Note: CC is correlation coefficient.)	115
4.23	Box plot of the measured FICD against various percentages of BARC over-etch trimming.	117

4.24	I_{dsat} vs. relative change in L_{eff} plot of wafer 1 (0 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	118
4.25	I_{dsat} vs. FICD plot of wafer 1 (0 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	119
4.26	I_{dsat} vs. relative change in L_{eff} plot of wafer 2 (25 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	119
4.27	I_{dsat} vs. FICD plot of wafer 2 (25 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	120
4.28	I_{dsat} vs. relative change in L_{eff} plot of wafer 3 (50 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	120
4.29	I_{dsat} vs. FICD plot of wafer 3 (50 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	121
4.30	I_{dsat} vs. relative change in L_{eff} plot of wafer 4 (75 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	121
4.31	I_{dsat} vs. FICD plot of wafer 4 (75 % over-etch) (Note: R^2 is the square of the correlation coefficient.)	122
4.32	I_{dsat} vs. relative change in L_{eff} plot for all the four wafers (Note: R^2 is the square of the correlation coefficient.)	123
4.33	I_{dsat} vs. FICD plot for all the four wafers (Note: R^2 is the square of the correlation coefficient.)	123
4.34	I_{dsat} vs. L_{eff} extracted using our Modified Shift-and-Ratio method for NMOS devices with different physical gate length. (Note: R^2 is the square of the correlation coefficient.)	124
4.35	MS&R L_{eff} vs. FICD measurement of NMOS devices with various physical gate lengths.	125
5.1	Plot of ΔL generated by Subthreshold Regime-Based method using V_{th} value extracted by different threshold voltage extraction methods with different channel length of n-MOS transistors fabricated using 0.11 μm CMOS technology.	131
5.2	Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.11 μm CMOS technology with L_{drawn} of 0.12 μm	134
5.3	Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.13 μm CMOS technology with L_{drawn}	135

of 0.12 μm , 0.13 μm and 0.15 μm .

- 5.4 Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.1 μm CMOS technology with L_{drawn} of 0.08 μm , 0.09 μm and 0.11 μm . 136
- 5.5 Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 90 nm CMOS technology with L_{gate} of 60 nm. 137

List of tables

Table		Page
4.1	A comparison of extracted L_{eff} values for NMOS between our modified method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.13 μm technology. L_{gate} is the physical gate length measured by XTEM	92
4.2	A comparison of extracted L_{eff} values for NMOS between our modified method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM	92
4.3	A comparison of extracted L_{eff} values for PMOS between our modified method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.13 μm technology. L_{gate} is the physical gate length measured by XTEM	93
4.4	A comparison of extracted L_{eff} values for PMOS between our modified method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM	93
4.5	A comparison of extracted L_{eff} values for NMOS transistors by 3 different methods, namely, MS&R, S&R, and SCM methods with various values of L_{drawn} for both 0.13 μm and 0.1 μm CMOS technology. L_{gate} is the physical gate length measured by XTEM.	102
4.6	A comparison of extracted L_{eff} values for PMOS transistors between MS&R, S&R and SCM measurement with different L_{drawn} for both 0.13 μm and 0.1 μm CMOS technology. L_{gate} is the physical gate length measured by XTEM.	102
4.7	A comparison of extracted L_{eff} values for NMOS transistors measured by MS&R, capacitance and S&R methods with various values of L_{drawn} for 0.11 μm technology. L_{gate} is the physical gate length measured by XTEM.	110
4.8	A comparison of extracted L_{eff} values for PMOS transistors between MS&R, capacitance and S&R method with different L_{drawn} for 0.11 μm CMOS technology. L_{gate} is the physical gate length measured by XTEM.	110
4.9	Inline poly DICD/FICD results (DICD and FICD were measured in μm .)	116

5.1	A comparison of threshold voltage values extracted by various methods for n-MOS transistor A with various channel lengths.	130
5.2	A comparison of threshold voltage extracted by various methods for n-MOS transistor B with various channel length.	130
5.3	A comparison of extracted L_{eff} values for NMOS transistors between Subthreshold Regime-Based, MS&R, capacitance and S&R method with different L_{drawn} for 0.11 μm technology. L_{gate} is the physical gate length measured by XTEM.	132
5.4	A comparison of extracted L_{eff} values for PMOS transistors between Subthreshold Regime-Based, MS&R, capacitance and S&R method with different L_{drawn} for 0.11 μm technology. L_{gate} is the physical gate length measured by XTEM.	133
5.5	A comparison of extracted L_{eff} values for NMOS transistors among Subthreshold Regime-Based, MS&R, S&R, and SCM measurement with different L_{drawn} for both 0.13 μm and 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM	133
5.6	A comparison of extracted L_{eff} values for PMOS transistors between Subthreshold Regime-Based, MS&R, S&R and SCM measurement with different L_{drawn} for both 0.13 μm and 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM.	133

CHAPTER ONE

Introduction

This thesis is a presentation of the candidate's research work performed as a PhD student of Nanyang Technological University attached to Chartered Semiconductor Manufacturing Ltd. This work involves the detailed study and understanding of physical limits and shortcoming of existing electrical parameter extraction methodology i.e. effective channel length (L_{eff}) extraction. This work not only improves and overcomes the limit of an existing L_{eff} extraction method. In fact, an original and novel method of L_{eff} extraction has been developed through the discovery of new device physics for state-of-the-art metal-oxide-semiconductor field-effect-transistors (MOSFETs) with ultra-thin gate oxide (≤ 2 nm). The following sections introduce the background and the purpose of this work while the details will be presented in the following chapters.

1.1 MOSFET scaling and device characterization challenges

Since 1958, the beginning of the integrated-circuit (IC) era, the complexity of integrated circuits has grown exponentially according to the Moore's law [1] from small-scale integration (SSI) to today's ultra-large-scale integration (ULSI) where the number of transistors reaches hundreds of millions in a chip as shown in Fig. 1.1 [2]. This sustained growth in ULSI technology is primarily driven by the continued shrinking of transistors to ever smaller dimensions.

Fig. 1.1: Exponential growth of the number of components per IC chip versus year for (a) Si-based MOS, (b) Si-based bipolar, (c) III-V semiconductor based MESFET, and (d) III-V semiconductor based MODFET [2].

This continued miniaturization in ULSI devices implies improvement of device switching speed, reduction of power consumption and overall device size. The smaller device size implies an increase in packing density, resulting in a reduction of unit price per function. As shown in Fig.1.2, one can see that the minimum feature length (i.e. the channel length of MOSFET) has been scaled down from 10 μm in 1970 to 0.18 μm in 2000 and based on the prediction by The International Technology Roadmap of

Semiconductors (ITRS) [3], device parameters will scale down from today's 90 nm CMOS technology to 10 nm CMOS technology in the year 2018.

Fig. 1.2 : Exponential decrease of (a) minimum feature length, (b) junction depth, and (c) gate oxide thickness of MOSFET [2].

According to scaling theory, both lateral dimensions, defined by lithographic feature sizes, and vertical dimensions such as junction depths (X_j) of a MOSFET, should be reduced. It involves all aspects of the MOSFET technology, such as lithography, etching, power supply voltage, threshold voltage, short-channel effects (SCE), gate dielectric thickness, high field effects, dopant density fluctuation, interconnect delay, device reliability, novel device structures, etc.[4,5]. However, this rapid scaling gives rise to major problems and issues, and these issues become more difficult to handle with newer technology generations [6,7]. In addition, as MOSFET is being aggressively scaled

down, more advanced device characterization and modeling techniques are required [8, 9].

1.2 Issues and limitation of current method of L_{eff} extraction with channel length scaling

Throughout the history of integrated circuit technology development, shrinking the dimensions of the transistor has required many difficult challenges to be overcome. It is not surprising that the current push towards sub-100 nm CMOS technology requires numerous innovations aimed at overcoming major challenges. Over the years, substantial research effort has been focused on developing a simple yet effective method to extract the effective channel length, L_{eff} [10]. This parameter contributes to the correlation between model and hardware, and is a key parameter to evaluate the performance of new generations of MOS transistors. Accurate extraction of the L_{eff} in MOS transistors is extremely important for both process monitoring/control and device/circuit design as devices are scaled down into the sub-100 nm regime. These parameters are not only necessary for circuit simulation and modeling, but are also needed as key input parameters to various current–voltage (I–V) characterization methods. Engineers use it to monitor lithography and etching during process development and manufacturing, and designers who model the I–V characteristics of a device need to know the L_{eff} of the devices they are modeling. Therefore, our ability to understand and characterize L_{eff} is important for the development of semiconductor technology.

In early works of L_{eff} extraction, the MOS transistors under test had a relatively simple source-drain (S/D) structure with an infinitely abrupt (laterally) S/D junction, as shown in Fig. 1.3 (a). For scaled MOS transistors, one of the key difficulties is the control

of short channel effect (*SCE*), which is the decrease of threshold voltage when there is a decrease in channel length due to two-dimensional electrostatic charge sharing between the gate and the *S/D* regions. One of the well-known approaches to suppress *SCE* is to use shallow source-to-well and drain-to-well junctions. However, the drain and source junctions cannot be too shallow because of the potential problem of junction spiking. Junction spiking occurs for both the old aluminum contact technology and also the newer silicide contact technology.

Fig 1.3: The evolution of MOSFET structures as shown in schematic cross-sections. (a) simple S/D MOSFET, (b) LDD MOSFET with relatively uniform channel doping, (c) MOSFET with punchthrough stopper used to suppress punchthrough and also SCE, (d) MOSFET with punchthrough stopper and also halo/pocket implant used to further suppress punchthrough and also SCE.

Fig. 1.3 shows the evolution of channel engineering of MOSFET structure in schematic cross sections as devices scaled down to sub-100 nm. MOSFET technology migrated from relatively early MOSFETs with a simple S/D structure to more advanced MOSFETs with shallow, lightly doped S/D extensions, formed by implantation of arsenic for n-channel and boron for p-channel MOS transistor. This is a structure called lightly doped drain (LDD) [11] which was devised to avoid electric field concentration near the drain thereby increasing the breakdown voltage and reliability against hot-carrier instability [12,13] of transistor. Back then, the semiconductor industry faced a major issue in extracting L_{eff} of MOSFET with LDD. Substantial effort had been put in to overcome this challenge [14-16]. In state-of-the-art sub-0.25 μm CMOS technology, the shallow S/D extension is no longer lightly doped but instead heavily doped such that the series resistance is smaller, and a deep conventional diffusion for the S/D is made to allow the use of silicide contact technology without junction spiking. In fact, this increase of LDD doping concentration helped to make the measurement of L_{eff} easier [17]. In addition, a halo or pocket implant is used as a punchthrough stopper [18,19] or for suppressing SCE (i.e. improving the threshold voltage V_{th} roll-off) [20-22]. This halo or pocket implant is particularly important for very small gate length transistors to control SCE as the MOS transistor is scaled down further. However, due to this implant, the short channel transistor features a degraded effective mobility compared to the long channel reference transistor [23]. This can be explained as shown in Fig 1.4: the effective channel doping increases if the halo regions overlap as channel length gets shorter. This results in a lower effective mobility for MOS transistors with a relatively smaller metallurgical channel length, thus L_{eff} extraction approaches based on the I-V characteristics at low drain bias (or linear region) tend to overestimate the value of L_{eff} . This is because for conventional

channel resistance L_{eff} extraction method, the effective mobility was assumed to be independent of channel length. Other approaches may involve algorithms which may be too complicated and difficult to implement in automatic wafer electrical testing systems for large volume of wafer testing.

Fig. 1.4: Schematic diagrams showing MOS transistors with halo implants (a) when the channel length is large and (b) when the channel length is small. It can be easily seen that for short MOS transistors, the halo implant from the source side overlaps with the halo implant from the drain side.

However, as MOSFETs are further scaled down, the accurate measurement of MOS transistor channel lengths will become increasingly difficult in the deep submicron region due to the fact that the surface mobility is not a constant but a function of the gate voltage. For the detailed fabrication process up to transistor level, please refer to Appendix A.

1.3 Motivation and objectives

As discussed in the earlier section, L_{eff} is an important parameter not only for circuit simulation and modeling, but also needed as a key input parameter for various current–voltage (I–V) characterization methods. Accurate L_{eff} extraction is a constant challenge due to the continuous scaling of CMOS technology. Therefore, our ability to understand and characterize L_{eff} is important for the development of semiconductor technology. This has motivated us to study and investigate the main effects on the accuracy of existing L_{eff} extraction methods resulting from MOSFET scaling. Hence one of the main objectives of this thesis is to develop a simple and efficient way of extracting L_{eff} accurately for routine process monitoring. First, we propose a modified and improved modified shift-and-ratio method, which is more suitable for MOS transistors fabricated by sub-0.25 μm CMOS technology. Secondly, we developed and verified a novel and original method suitable for advanced state-of-the-art CMOS technology based on discovery of new device physics.

1.4 Organization and contribution of the thesis

This thesis consists of five chapters and its organization is as follows: Chapter One presents the brief history of CMOS technology trend, the challenges face in extracting L_{eff} as MOSFET scaled down. It also elaborates on the issues and limitation of L_{eff} extraction method when applied to today’s state-of-the-art MOS transistors with halo/pocket implant.

Chapter Two then gives a review of two major types of L_{eff} characterization, namely electrical and physical characterization. This provides a good understanding about different L_{eff} definitions and extraction methods. In addition, detailed physics, limitation, advantages and drawback of each extraction method are discussed. Following that, Chapter Three describes the device fabrication and experimental setup. In addition, it also presents the proposed physical theory of our L_{eff} extraction method. Furthermore, various approaches to overcome existing limitations of extracting L_{eff} are presented. In Chapter Four, the extracted L_{eff} generated by the Modified Shift-and-Ratio method are shown to be accurate by applying the new techniques on advanced 0.13 μm , 0.11 μm and 0.1 μm MOS transistors. Comparison of the extracted L_{eff} was made to that measured by Scanning Capacitance Microscopy (SCM) and Capacitance L_{eff} extraction method. In Chapter Five, the novel Subthreshold Regime-based method was successfully applied on advanced 0.13 μm , 0.11 μm and 0.1 μm MOS transistors. This demonstrates the usefulness and capability of the new method in extracting L_{eff} successfully on advanced MOS transistors with halo/pocket implant. The last chapter concludes the thesis and offers suggestions for future research direction based on the results of this thesis. Appendix A provides the schematic cross-section of a typical CMOS process flow up to transistor level and the details of the poly etch process. Appendix B gives an overview of some of the threshold voltage extraction methods available in the industry; in addition, the source code of different threshold voltage extraction algorithms was included.

CHAPTER TWO

Review of L_{eff} Extraction Methods and Effective Mobility for MOSFET

2.1 Introduction: Semiconductor device characterization

All semiconductor devices and materials are required to be characterized to a certain extent, be it through the use of test structures together/or with wide spectrum of measurement methods found in the modern semiconductor industry.

Semiconductor material and device characterization techniques can generally be divided into (1) electrical characterization, (2) optical characterization (e.g. Raman spectroscopy) and (3) chemical and physical characterization (e.g. electron beam, ion, X-ray, probe method etc.). Electrical parameters of a device include series resistance, effective channel length, L_{eff} , threshold voltage, mobility etc. One of the most basic and important MOSFET parameters is the L_{eff} . The L_{eff} is used as a starting point to MOSFETs characterization, simulations and modeling as it is a direct measure of the process controllability and reproducibility. As understanding of device physics has become even more important now that MOSFETs have crossed the long channel frontier into the sub-micron realm. Many device effects observed in sub-micron MOSFETs are impacted by the process technology used to fabricate them. Only with such an understanding can the relationship between circuit behavior, device design, and process technology of sub-micron MOSFETs be grasped. Consequently, by gaining such an understanding process

engineers will be better able to contribute to the task of successfully designing and manufacturing sub-micron ICs.

The following section describes the various definition of L_{eff} . The subsequent sections in this chapter review the different methodology in characterize channel length, both electrically and physically. Physical technique such as scanning electron microscopy (SEM), transmission electron microscopy (TEM) and scanning capacitance microscopy (SCM) are briefly presented. The advantages and drawback of each technique are discussed as well.

2.2 Various definitions of channel length

A number of terms have been used to describe the length of a MOS transistor. Some common terms include, mask length (L_{mask}), gate length (L_{gate}), metallurgical channel length (L_{met}), and effective channel length (L_{eff}).

Fig. 2.1 in the following page shows schematically how various channel lengths are defined. L_{mask} is the design length on the polysilicon etch mask, sometime it is also known as the drawn length of the polysilicon, L_{drawn} . L_{gate} is reproduced on the wafer after the mask pattern is transfer to the wafer through lithography and etching processes. L_{gate} can be either longer or shorter than L_{mask} depend on the lithography and etching biases. For the same L_{mask} design, L_{gate} may vary from die to die, wafer to wafer, and lot to lot. Although L_{gate} is an important parameter for process monitoring and control, there is no simple way of making a large quantity of measurements. Typically, L_{gate} is measured by CD-SEM (a specialized scanning electron microscope for critical dimension measurement) routinely in the production line. However, this is done for only a limited number of wafers in the

production line. Cross-sectional transmission electron microscopy (XTEM) can be used as well but it is not easy to prepare sample for XTEM.

Fig. 2.1: Schematic diagram showing the relationship among the various definitions of channel length [24].

L_{met} is defined as the distance between the metallurgical junctions of the source and drain diffusions at the silicon surface. Since the source and drain regions are self-aligned to the polysilicon gate in a standard process, there is a close correlation between L_{met} and L_{gate} . Usually, L_{gate} is longer than L_{met} by a certain amount due to the lateral straggle of ion implantation and the lateral source-drain diffusion in the process. Accurate physical measurement of L_{met} is very difficult because the drain and source junctions cannot be easily seen. With specially prepared sample, L_{met} may be measured using cross-sectional SCM. Furthermore, using specially prepared sample cross-section with stained drain and source junctions, L_{met} may be measured using XTEM. (Note: the junction staining may be difficult.) However, these methods are obviously destructive.

The parameter L_{eff} is different from all other channel lengths discussed above in that it is defined through some electrical characteristics of the MOS transistor device and as such is not strictly a physical parameter. A proper physical interpretation of the extracted L_{eff} is in terms of the injection points where the MOS transistor current spreads from the surface layer into the bulk source-drain region. Qualitatively, L_{eff} is a measure of how much gate-controlled current a MOS transistor delivers in reference to the long-channel device.

2.3 Electrical characterization- Extraction of L_{eff}

There are basically four different types of approach in extracting L_{eff} electrically, one of the most common approaches is based on the device's I-V characteristics at low drain bias or linear region [14,15,25-40], the second type of approach is base on the measured C – V data of a series of MOS transistors with different channel lengths [16,41-48]. Recently, Ye et al. [49] proposed an entirely different approach based on drain-induced-

barrier-lowering (DIBL) effect to determine L_{eff} . Some other approaches include extraction of L_{eff} based on measurement of saturation voltage [50]; “Critical-Current at Linear-Threshold” [51]; direct tunneling current to extract L_{eff} . [52,53]. This approach is more complicated because it is necessary to calibrate all modeling parameters for a given gate oxide thickness and needs to take into account non-uniform gate oxide thickness.

2.3.1 Channel Resistance based method

One of the largest families of L_{eff} extraction approach is based on the device’s $I-V$ characteristics at low drain bias or linear region. The very basic assumption of this approach is that the channel resistance of a MOSFET in the linear or low-drain bias region is proportional to the channel length. From the device’s $I-V$ characteristics, different assumptions were made use of in developing different methods to extract L_{eff} [28].

Ng & Brews gave an excellent review on the various methods proposed to measure L_{eff} up to 1990 [28]. Most of the extraction methods are based on the device’s $I-V$ characteristics at low drain bias or linear region, $V_{ds} < (V_{gs} - V_{th})$. The drain current in the linear region of MOS transistors is given by

$$I_{ds} = \mathbf{m}_{eff}(V_{gs}) C_{ox} (W_{eff} / L_{eff}) (V_{gs} - V_{th}) V_{ds} \quad (2.1)$$

where I_{ds} and V_{ds} are the source-drain current and voltage respectively, V_{gs} is the gate-source voltage, V_{th} is the linear threshold voltage, C_{ox} is the gate oxide capacitance per unit area, W_{eff} is the effective gate width, $\mathbf{m}_{eff}(V_{gs})$ is the effective mobility which is a weak function of V_{gs} and L_{eff} is the effective channel length.

The second-order term $-V_{ds}/2$ in the $(V_{gs}-V_{th})$ factor in Eq. (2.1) is omitted for simplicity (which makes no difference for some of the L_{eff} extraction method). For different L_{mask} , L_{eff} differs but is assumed to relate to L_{mask} by a constant (independent of bias voltages) ΔL .

$$L_{eff} = L_{mask} - \Delta L \quad (2.2)$$

All the lithography and etch biases as well as the lateral S/D implant straggle and diffusion are lumped into ΔL . The basis of L_{eff} definition lies in the fact that the channel resistance of a MOS transistor in the linear or low-drain bias region is proportional to the channel length. That is

$$R_{ch} \equiv \frac{V_{ds}}{I_{ds}} = \frac{L_{eff}}{\mu_{eff}(V_{gs})C_{ox}W_{eff}(V_{gs}-V_{th})} \quad (2.3)$$

In the presence of series resistance from the source side R_s , and the drain side R_d , as shown in the equivalent circuit in Fig. 2.2, R_s and R_d are assumed to connect an *intrinsic* MOS transistor to the external terminals where a voltage is applied between the source and drain. If one defines the total source/drain parasitic resistance as $R_{sd} = R_s + R_d$, the externally measured total resistance can be written as shown below:

$$\begin{aligned} R_{tot} &\equiv \frac{V'_{ds}}{I_{ds}} \\ &= R_{sd} + R_{ch} \end{aligned}$$

$$= R_{sd} + \frac{L_{mask} - \Delta L}{m_{eff} C_{ox} W_{eff} (V_{gs} - V_{th})} \quad (2.4)$$

Here, we do not distinguish between the intrinsic and the external V_g since the effective V_g varies only slightly from the applied gate voltage because $V_{ds} < (V_g - V_{th})$. A key assumption in almost all L_{eff} extraction algorithms is that, in contrast to R_{ch} , R_{sd} is independent of gate voltage. In other words, R_{sd} is defined as the constant part of the measured device resistance, consistent with existing circuit models.

Fig. 2.2: Schematic diagram of MOS transistor with S/D series resistance [24].

Almost all the channel resistance methods of L_{eff} extraction begin with Eq. (2.4) and need at least two different devices with different channel length to deduce L_{eff} . From here, they go their separate ways and make use of different assumptions. One of the most commonly used methods of L_{eff} extraction [26], called the *channel resistance method* will be discussed in the following section. This will be followed by the Shift-and-Ratio method [29], which has been considered one of the most accurate and consistent techniques to extract the L_{eff} .

2.3.1.1 Channel resistance method by Chern et al. published in 1980

Fig. 2.3: L_{eff} extraction method modified from the original method proposed by Chern et al. [26]

One of the most commonly used methods of L_{eff} extraction is the *channel resistance method* due to its simplicity. From Eq. (2.4), the total measured resistance is the sum of the channel resistance and series resistance, which is proportional to L_{eff} . For a fixed V_{gs} , it is assumed that $\mu_{eff}(V_{gs})$ is constant for different channel lengths with $\partial R_{sd} / \partial V_{gs} = 0$ and R_{tot} is linearly dependent on L_{mask} . For a set of devices with different L_{mask} but the same W_{eff} , the parameters R_{sd} , DL , and C_{ox} are the same within process tolerances. The linear threshold voltage V_{th} , however, does depend on channel length because of the short channel effect (SCE). When comparing R_{tot} of devices with different L_{mask} , it is important to measure V_{th} for each device and adjust such that the gate overdrive ($V_{gs} - V_{th}$) is the same from device to device. A plot of R_{tot} (at small V'_{ds}) at different ($V_{gs} - V_{th}$) versus at least two different L_{mask} would then yield straight lines that passes through the point (DL , R_{sd}) as shown in Fig. 2.3. The slope of the line depends on the specific value of the gate

overdrive. The common intersection of the lines for different $(V_{gs} - V_{th})$ determines both DL and R_{sd} . However, R_{tot} versus L_{mask} lines for different gate overdrives may not intersect at a common point. Significant errors may result if only a limited number of $(V_{gs} - V_{th})$ are investigated. The problem of measuring R_{tot} accurately become more serious as MOS transistor is scaled down into sub-micron region due to strong variation of mobility with gate voltage, more pronounced effects of parasitic source-drain resistance, and linewidth-dependent lithography bias near optical resolution limit.

2.3.1.2 Shift-and-Ratio method proposed by Taur et al. [29]

An improved channel length extraction algorithm, called the “shift-and-ratio” method [29], is able to circumvent the problems as mention in the last section. It allows the channel mobility to be an arbitrary function of gate voltage and, at the same time, is able to determine the threshold voltage of short-channel transistor independent of parasitic series resistance. The shift-and-ratio method is based on the same channel resistance concept described above. It starts with a generalization of Eq. (2.4) to the following form:

$$R_{tot}(V_{gs}) = R_{sd} + L_{eff} f(V_{gs} - V_{th}) \quad (2.5)$$

where f is a general function of gate overdrive $(V_{gs} - V_{th})$ common to all the measured devices independent of the gate length. The key assumption behind Eq. (2.5) is that the effective mobility μ_{eff} is a common function of $(V_{gs} - V_{th})$ for all the measured devices independent of the gate length.

To solve for L_{eff} in Eq. (2.5) from the measured data of R_{tot} , we can define $S(V_{gs})$ as the derivative of R_{tot} with respect to V_{gs} . Since the parasitic resistance is independent or a weak function of V_{gs} , its derivative can be neglected:

$$S(V_{gs}) \equiv \partial R_{tot} / \partial V_{gs} = L_{eff} \frac{df(V_{gs} - V_{th})}{dV_{gs}} \quad (2.6)$$

Here, df/dV_{gs} is also a general function of gate overdrive common to all the devices under test. Fig. 2.4 a shows an example of $S(V_{gs})$ both long channel and short channel devices. In general, L_{eff} extraction is done by considering a short channel MOS transistor with $L_{eff}=L_i$ and a long channel MOS transistor with $L_{eff}=L_o$ which serves as the reference device. The ratio L_o/L_i can be shown to be equal to the ratio $r(V_{gs})$ of the derivative $S_o(V_{gs})$ for the reference MOS transistor and the derivative $S_i(V_{gs} + \Delta V_{th})$ for the MOS transistor under test with the same gate width but shorter channel length as follows:

$$r(V_{gs}) = S_o(V_{gs}) / S_i(V_{gs} + \Delta V_{th}) = L_o / L_i \quad (2.7)$$

$$\text{where } \Delta V_{th} = V_{tho} - V_{thi}$$

$$L_{eff} = L_i = L_{drawn} - \Delta L \quad (2.8)$$

Subscripts o and i refer to the reference device and the specific device under test respectively. R_{sd} is assumed to be constant with gate bias, and so it can be conveniently removed in the derivatives. In addition, it is assumed that the mobility is a function of $(V_{gs}-V_{th})$, i.e., $\mu(V_{gs}-V_{th})$ is common for the two devices. In order to solve L_{eff} , ΔV_{th} is adjusted until $r(V_{gs})$ becomes a constant function of V_{gs} . Once a constant ratio of $r(V_{gs})$ is achieved, L_{eff} is calculated using Eq. (2.7). The procedure of finding constant ratio of

$r(V_{gs})$ can be automated by computing the average $r(V_{gs})$ and the mean square deviation of $r(V_{gs})$ from its average value, i.e.,

Fig. 2.4: (a) $S(V_{gs})$ curves for long-channel and short-channel devices. The device width is 10 μm . (b) Average ratio (open circle) and variance (solid dot) versus gate voltage shift [29].

$$\langle r \rangle = \frac{\int_{\Delta V_{gs}} r(\mathbf{d}, V_{gs}) dV_{gs}}{\int_{\Delta V_{gs}} dV_g} \quad (2.9)$$

$$\text{and} \quad \langle \mathbf{s}^2 \rangle = \langle r^2 \rangle - \langle r \rangle^2 \quad (2.10)$$

as a function of ΔV_{th} for selected range of gate voltage. Fig. 2.4b shows that the $\langle s^2 \rangle$ versus V_g shift curve exhibits a sharp minimum at the best match (constant ratio).

2.3.1.3 Limitation of current *Channel Resistance* method of L_{eff} extraction

In early works of L_{eff} extraction, the MOS transistors under test had a simple S/D structure with an infinitely abrupt (laterally) S/D junction while the pocket implant is absent. However, as devices are being scaled down. The older methods are not efficient in extracting L_{eff} for deep sub-micron MOS transistors. One of the most popular techniques for extracting L_{eff} in deep sub-micron MOS transistors is the “shift-and-ratio” method proposed by Taur et al [29,54], which has been considered as one of the most accurate and consistent techniques to extract the L_{eff} . Various modifications of the original shift-and-ratio technique [30,35,38,55] have also been proposed. However, when Taur et al. proposed their method in 1992, they considered a MOS transistor with a simple source/drain (S/D) structure. In fact, Taur et al. pointed out that their method is not suitable for MOS transistors with a shallow LDD (light doped drain) and a deep conventional diffusion for the S/D [24,54]. Since the carrier concentration and, thus the resistance in the lightly-doped regions are strongly dependent on gate bias, the LDD structure make the channel length definition ambiguous. L_{eff} extraction methods suitable for MOS transistors with LDD had been proposed [14-16], but will not be discussed in detail in this chapter. Nowadays, the drain/source of sub-0.25 μm CMOS technology has

a shallow S/D extension, which is no longer lightly doped, and a deep conventional diffusion for the S/D. In addition, a halo or pocket implant is used as a punchthrough stopper [56] or for improving the threshold voltage V_{th} roll-off [57-58]. This halo or pocket implant is particularly important for very small gate length transistors. However, due to this implant, the short channel transistor features a degraded effective mobility compared to the long channel reference transistor [23]. It was recently reported by van Meer et al. in 2000 that there are some basic limitations of the “shift-and-ratio” technique for state-of-the-art MOS transistors with halo or pocket implants such that L_{eff} tends to be significantly over-estimated [23]. Our own measurement base on the “shift-and-ratio” method shows consistent results as reported in the literature when this technique was applied to sub-0.25 μm CMOS transistors, the measured L_{eff} result will be presented in Chapter Four. This limitation applied to almost all L_{eff} extraction method based on the *channel resistance* method with the assumption that the mobility is independent of the gate length [59]. Fig. 2.5 is a clear illustration that the total channel resistance of devices with L_{mask} smaller than 0.25 μm is substantially higher than predicted by linear scaling from longer devices. The higher channel doping concentration of short channel devices, leads to stronger Coulomb scattering (particularly affecting mobility at low effective fields) and to a higher vertical field for the same gate overdrive compared to long channel devices. Both these effects contribute to a mobility reduction, hence to an increased resistivity. Thus, L_{eff} tends to be significantly over-estimated as channel resistance is no longer independence of channel length as technology scaled down to sub-0.25 μm regime.

Fig. 2.5: Total resistance $R_{TOT} = V_{DS}/I_{DS}$ of scaled devices biased at the same gate overdrive $V_{GT} = (V_{GS} - V_T)$: Dashed lines represent the linear regression of R_{TOT} (at the three different values of V_{GT}), calculated considering only the data measured on MOSFET's with L_{mask} ranging from 0.35 to 0.9 μm [59].

2.3.2 Capacitance based method

The second type of approach is based on the measured C-V data of a series of MOS transistors with different channel lengths [16,41-43,46-48]. It is based on the fact that, when a MOS transistor is turned on, the intrinsic inversion gate-to-channel capacitance (C_{gc}) is proportional to the channel length. Hsieh et al. [46] proposed a similar C-V extraction method based on accumulated gate-to-channel capacitance (C_{gb}) suitable for MOS devices with pocket implants in which L_{eff} can be extracted from two devices with different gate lengths. These approaches are more complicated because of the small capacitance value to be measured and the non-negligible parasitic capacitance. [60-62]

The total measured gate-to-channel capacitance $C_{gctotal}$ is the sum of the parasitic C_p , the overlap, C_{ov} and C_{inov} , and, the intrinsic channel capacitances C_{gc0} where $C_{ov} = C_{ov1} + C_{ov2}$ and is expressed as [62]

$$C_{gctotal}(V_g, L) = C_p + C_{ov} + C_{inov}(V_g) + C_{gc0}(V_g, L) \quad (2.11)$$

After subtraction of the total parasitic capacitance $C_p + C_{ov}$ from (2.11), where $C_p + C_{ov}$ was measured at strong accumulation. The gate-to-channel capacitance corresponds therefore to

$$C_{gc}(V_g, L) = C_{inov}(V_g) + C_{gc0}(V_g, L) \quad (2.12)$$

The component $C_{gc}(V_g)$ is nearly length independent appearing below threshold stems from the gate voltage dependent inner overlap capacitance C_{inov} , coupling the polygate and the source-drain junctions through the silicon substrate in depletion regime. This C_{inov} component is fully screened out when going into strong inversion or accumulation [62]. Therefore in strong inversion, the maximum values, C_{gcmax} are not affected by the inner overlap capacitances; one can extract the effective gate length from $C_{gcmax}(L_{drawn})$ data as

$$L_{eff} = L_s = \left[\frac{C_{gcmax,s}}{C_{gcmax,l}} \right] \cdot L_l \quad (2.13)$$

where L_s refer to short channel length and L_l long channel length (e.g. $L_l = 10\mu\text{m}$)

Fig. 2.6 shows the $C_{gc}(V_g) = C_{inov}(V_g) + C_{gc0}(V_g, L)$ characteristics as obtained on nMOSFETs of various gate lengths after subtraction of the total parasitic capacitance $C_p + C_{ov}$ from (2.11). $C_p + C_{ov}$ was measured in strong accumulation for each gate length.

Fig. 2.6: Simulated $C_{dg} + C_{sg}$ versus V_g for n-MOSFETs with different gate lengths. This plot illustrates the separation between intrinsic and extrinsic regions. In particular the extrinsic capacitance is shown to be bias dependent. Inset: MOSFET structure generated with the 2D process simulator [62].

Although several C-V methods have been proposed to avoid the difficulties of extracting L_{eff} using I-V methods for LDD MOSFETs and advanced MOS transistors with halo/pocket implants, there are some setbacks for the C-V methods as follows. For example, an additional test device with a large gate width is required to measure the inversion gate-to-channel capacitance (C_{gc}) accurately, because of the small value of C_{gc} in real short-channel MOSFETs. In this case, the inconsistency of the test structure from real devices may generate severe extraction error. To avoid the error, it is better to use real devices directly. However, it is very difficult to measure C_{gc} using conventional C-V measurements which are not sensitive enough to determine low C_{gc} values in ultra-short-channel devices. In addition the accurate procedure to eliminate overlap and fringe

capacitances from the measured C_{gc} should be performed. This elimination process becomes very important for deep-submicron MOSFETs, because the influence of parasitic capacitance on C_{gc} becomes more dominant for shorter L_{eff} . However, there is no direct method to measure outer fringing capacitance, C_{of} . Conventionally, C_{of} is used as a fitting parameter calibrated with the gate length measured by SEM or TEM. Furthermore, for sub-100 nm MOSFET, the CV measurement on ultra-thin gate oxide ($T_{ox} < 2$ nm) must account for the parasitic resistance due to strong direct tunnel current [63].

2.3.3 DIBL method

Recently, Ye et al. [49] proposed an entirely different approach based on the drain-induced-barrier-lowering (DIBL) effect to determine L_{eff} . DIBL is a primary indication of SCE: as L_{eff} decreases, the potential barrier between source and channel is lowered by the presence of a voltage applied to the drain, hence less gate voltage is needed to turn on the channel, resulting in a smaller V_{th} . The widely accepted DIBL model shows that the shift between V_{th} measured at low and high V_{DS} (ΔV_{th}) is exponentially dependent on L_{eff} [64].

$$\Delta V_{th}^{dibl}(L) = [3(V_{bi} - 2\phi_B) + V_{ds}]e^{-L_{eff}/l_c} + 2\sqrt{(V_{bi} - 2\phi_B)(V_{bi} - 2\phi_B + V_{ds})}e^{-L_{eff}/2l_c} \quad (2.14)$$

Eq. (2.14) can be simplified as follows:

$$\Delta V_{th}^{dibl}(L) = a + b \exp\left(-\frac{L_{eff}}{2l_c}\right) \quad (2.15)$$

(for : $0.5 \mu\text{m} > L_{eff} > 2l_c$)

The exponential dependence of $\Delta V_{t_{dibl}}$ on L_{eff} agrees with experimental results very well as shown in Fig. 2.7 where $\Delta V_{t_{dibl}}$ is plotted versus $L_{poly(cap)}$. Eq. (2.16) implies the possibility of L_{eff} extraction based on DIBL effect.

$$L_{eff(dibl)} = -2l_c * \ln \left(-\frac{\Delta V_{t_{dibl}} - a}{b} \right) \quad (2.16)$$

(for : $0.5 \mu m > L_{eff} > 2l_c$)

Fig. 2.7: Example of $\Delta V_{t_{dibl}}$ plotted versus $L_{poly(cap)}$ for NMOSFET devices. The trend line shows the exponential dependence of $\Delta V_{t_{dibl}}$ on $L_{poly(cap)}$ [49].

2.3.4 Direct tunneling method

Another type of approach utilizes the direct tunneling current through thin gate oxide for obtaining the L_{eff} . These approaches are insensitive to doping concentration and

are free from the assumption of unified effective mobility between transistors with different channel length. Techniques making use of the direct tunneling current include the Direct-Tunneling-Current (DTC) method proposed by Hong et al. [52], similar technique, using the gate tunneling current measurements from accumulation to strong inversion proposed by Martin et al. [53] and van Langevelde et al. [65]

Fig 2.8: Schematic view of direct tunneling current (DTC) of a MOSFET.

As shown in Fig. 2.8, the direct tunneling current take place at the channel (L_{met}) and extension (L_{ext}) regions. The gate-to-channel and gate-to-bulk component of the current are related to the metallurgical gate length while the gate-to-source/drain current is to the S/D extension length. For different channel length, the S/D extension overlap length from the junction to gate edge is the same while the metallurgical channel length follows the poly gate length [66]. Fig. 2.9 shows plots of the direct tunneling current for different channel length. At low bias condition, the $I_{gs} + I_{gd}$ (gate-to-source/drain) component of the direct tunneling current mostly arises from the S/D overlap regions. As the gate voltage increase towards the channel inversion and accumulation region, the

direct tunneling current dominantly comes from the channel region. Therefore, if the direct tunneling current densities of the channel and the S/D extension regions are known, the physical gate length can be easily obtain. The direct tunneling current between the gate-to-channel region need to be measured on a large MOSFET transistor as a reference device. However, the current density in the S/D extension region can only be obtained through simulation-base virtual S/D extension MOSFET device. The channel as well as the S/D extension length will then be extracted by comparing the direct tunneling current of the short channel device with the direct tunneling current density of the reference and simulated device.

Fig 2.9: Measured and modeled direct tunneling current at $V_{ds}=0.05\text{V}$ at different channel length. [66]

Although these approaches are insensitive to doping concentration and are free from the assumption of unified effective mobility between transistors with different channel length but, these approaches required a gate current model or simulation and some additional parameter extraction in some cases. In addition, gate oxide non-uniformity, gate oxide to silicon interface roughness, interface states or trapped charge could potentially cause small discrepancy in the direct tunneling current measurement.

2.4 Physical characterization

Besides characterizing the channel length electrically, physical characterization is sometimes used as a complementary technique for cross-sectional analysis to determine device dimensions. As mentioned in Section 2.2, the physical gate length, L_{gate} of a MOSFET can be characterized by using SEM or XTEM [67]. However, these techniques are generally more specialized and require more complicated and thus more expensive equipment than those used for electrical characterization. In addition, the main drawback of these techniques is that they are destructive.

Recently, a new technique has been used to obtain two-dimensional dopant profiles of semiconductor devices by scanning capacitance microscopy (SCM) [68-70]. For the case of MOSFETs, the SCM image shows high contrast at the differently doped areas such as source, drain and gate. The SCM image can be used to extract valuable information such as the L_{eff} , junction depth, or details on the lateral and vertical extension of the doped regions. The following sections give a brief introduction of SCM, SEM and TEM.

2.4.1 Scanning capacitance microscopy

Scanning capacitance microscopy (SCM) can be used to show two-dimensional carrier concentration of semiconductor materials and simultaneously acquire AFM images of critical device structures with high resolution. It is a member of Scanning Probe Microscopy (SPM) techniques which employ the contact mode of Atomic Force Microscopy (AFM). Contact mode probes of AFM consist of cantilevers with low spring constants ($< 1 \text{ N/m}$) to minimize the force between the tip and the sample during imaging as shown in Fig. 2.10. During scanning, the probe first touches the sample, resulting in bending of the cantilever. Then this bending is detected optically by a laser beam, which is reflected off the cantilever onto a photodetector. The photodetector gives an output and when this change of output reaches a certain level, the scanner stops when approaching to the surface.

Fig. 2.10: Optical lever of AFM [71].

At this position the cantilever will apply a constant force to the probe to scan the surface. An extremely precise positioning device made of piezoelectric ceramics, which we refer as scanner, performs the movement of the tip. When the probe encounters the topographic variation, the cantilever further will bend causing the output of the photodetector to change. The output of the photodetector is fed to a comparator. If it is determined by the comparator that the force on the cantilever has increased, the cantilever is pulled back from the surface until the photodetector output returns to the original level. By using a feedback loop, the tip can be adjusted to maintain a constant force while scanning across the surface. The tip height is recorded by a computer and displayed on screen as the topography image of sample surface. Atomic scale resolution can be achieved with a well designed AFM.

The very basis of SCM operation is the presence of an ultra high frequency (UHF) resonant capacitance sensor as shown in Fig. 2.11. The resonator is connected to the cantilever/tip assembly (with metal-coated tip) via a transmission line. When the resonating probe tip is put in contact with a semiconductor, the sensor, transmission line, probe and carriers in the sample near the tip all become part of the resonator. This means that the tip-sample capacitance and variations in it will load the end of the transmission line and change the resonant frequency of the system. Small changes in resonant frequency create enormous changes in the resonance sensor output signal. This system has been shown to be sensitive to variations as small as attofarads (10^{-18} F).

Fig. 2.11: SCM detection system schematic.

The operation of SCM is based on metal-oxide-semiconductor (MOS) capacitor theory. It works by exploiting the voltage dependent capacitance formed between the SCM probe tip and the semiconductor sample. The tip-sample junction is basically a MOS capacitor. This MOS capacitance changes with voltage since the free carriers in the semiconductor beneath the SCM probe tip are either attracted or repelled due to an alternating electric field, which is caused by an AC electric field coupled with a DC voltage. Taking the n type material as example, when a positive voltage is applied, the electrons are attracted the semiconductor/oxide interface and accumulate there. In the accumulation region, the total capacitance is just the parallel plate oxide capacitance between the metal and semiconductor. As the voltage bias applied becomes negative, the electrons move away from the semiconductor/oxide interface, thereby depleting the amount of carriers in that region, which is equal to increase the spacing between the equivalent capacitor plates, hence the capacitance decreasing. The high frequency

capacitance vs. voltage relationship for both high and low dopant concentration of n type semiconductor material is shown in Fig. 2.12.

Fig. 2.12: Capacitance vs. voltage curve for n-type semiconductor material.

As shown in Fig. 2.12, lower carrier concentration semiconductor deplete deeper, thus the capacitance decreases faster with respect to decreasing voltage. Since holes are the majority carriers for p-type semiconductor materials, the polarity of C-V curve is inverted and it displays accumulation at negative voltages and depletion at positive voltages.

As mentioned before, this tip-oxide-semiconductor structure is similar to a MOS capacitor. A MOS capacitor is formed by a metal electrode, an oxide layer and a semiconductor substrate. A typical MOS capacitor can be biased into any of four conditions: flatband condition, accumulation condition, depletion condition and inversion condition. The capacitance depends on the frequency of the AC bias voltage. For the quasi-static (low frequency) case, an inversion layer is formed as soon as the surface potential is equal to double bulk potential. The capacitance rapidly increases back to oxide capacitance value. If the frequency of the applied AC signal is high enough, the inversion charge (minority carriers) will not be able to follow the signal and only the

depletion charge (majority carriers) can respond. The high frequency capacitance is a constant value. If the AC bias is ramped up very fast (ramp time < minority carrier response time), the inversion layer will not have a chance to form. And the MOS capacitor will be biased into deep depletion. The depletion width will continue to increase and the capacitance will continue to decrease below the constant value. Most SCM measurements are made at this deep depletion condition.

Fig. 2.13: Change from accumulation to depletion during SCM due to alternating electric field.

To produce the variation in capacitance, which is actually measured in SCM, a depletion region is generated by applying a sinusoidal voltage to the tip or sample as shown in Fig. 2.13. This sinusoidal voltage has amplitude as $|V_{AC}|$ and frequency of 10-100 Hz. In the depletion condition, the major carriers are repelled from the surface. At two different voltage, $V_{DC} + |V_{AC}|$ and $V_{DC} - |V_{AC}|$, the depletion depths are different, causing variation of tip-sample capacitance. This variation is measured by the capacitance sensor at the AC bias voltage frequency as $dC = C(V_{DC} + |V_{AC}|) - C(V_{DC} - |V_{AC}|)$. V_{DC} is the DC voltage applied to the tip. The depth of depletion is also determined by the quality and thickness of the dielectric layer (usually an oxide) and the free carrier concentration.

Under thinner oxide layer, depletion will be deeper. If a sample has both high and low carrier concentration regions, the depletion in the low-concentration region would be deeper than the high-concentration region under the same applied AC voltage and same oxide thickness, leading to high capacitance variation and high SCM signal.

2.4.2 Scanning electron microscopy

An electron microscope utilizes an electron beam to produce a magnified image of the sample. In the scanning and transmission electron microscope, an electron beam incident on the sample produces an image. The SEM uses electrons instead of light to form an image. The use of electrons has two main advantages over optical microscopes: much larger magnifications are possible since electron wavelengths are much smaller than photon wavelengths and the depth of field is much higher. A beam of electrons is produced at the top of the microscope by heating of a metallic filament. The electron beam follows a vertical path through the column of the microscope. It makes its way through electromagnetic lenses which focus and direct the beam down towards the sample. Once it hits the sample, other electrons (backscattered or secondary) are ejected from the sample. Detectors collect the secondary or backscattered electrons, and convert them to a signal that is sent to a viewing screen producing an image. A schematic representation of an SEM is shown in Fig. 2.14.

In a typical SEM, electrons are thermionically emitted from a tungsten or lanthanum hexaboride (LaB_6) cathode and are accelerated towards an anode; alternatively, electrons can be emitted via field emission (FE). Tungsten is used because it has the highest melting point and lowest vapour pressure of all metals, thereby allowing it to be heated for electron emission. The electron beam, which typically has an energy ranging from a few hundred eV to 100 keV, is focused by one or two condenser lenses

into a beam with a very fine focal spot sized 1 nm to 5 nm. The beam passes through pairs of scanning coils in the objective lens, which deflect the beam horizontally and vertically so that it scans in a raster fashion over a rectangular area of the sample surface. When the primary electron beam interacts with the sample, the electrons lose energy by repeated scattering and absorption within a teardrop-shaped volume of the specimen known as the interaction volume, which extends from less than 100 nm to around 5 μm into the surface. The size of the interaction volume depends on the beam accelerating voltage, the atomic number of the specimen and the specimen's density. The energy exchange between the electron beam and the sample results in the emission of electrons and electromagnetic radiation which can be detected to produce an image.

The most common imaging mode monitors low energy (<50 eV) secondary electrons. Due to their low energy, these electrons originate within a few nanometers from the surface. The electrons are detected by a scintillator-photomultiplier device and the resulting signal is rendered into a two-dimensional intensity distribution that can be viewed and saved as a Digital image. This process relies on a raster-scanned primary beam. The brightness of the signal depends on the number of secondary electrons reaching the detector. If the beam enters the sample perpendicular to the surface, then the activated region is uniform about the axis of the beam and a certain number of electrons "escape" from within the sample. As the angle of incidence increases, the "escape" distance of one side of the beam will decrease, and more secondary electrons will be emitted. Thus steep surfaces and edges tend to be brighter than flat surfaces, which results in images with a well-defined, three-dimensional appearance. Using this technique, resolutions less than 1 nm are possible. Because secondary electrons are emitted from the specimen in an omni directional manner and possess relatively low energies they must be

in some way collected before they can be counted by the secondary electron detector. For this reason the secondary electron detector is surrounded by a positively charged anode or Faraday cup or cage that has a potential charge on it in the neighborhood of 200 V.

Fig. 2.14: Schematic of a scanning electron microscope

This tends to draw in many of the secondary electrons towards the scintillator. It also helps to alleviate some of the negative effects of the scintillator aluminum layer bias which because it is so much greater (10 KV vs. 200 V) can actually distort the incident beam. A second type of electron, the backscattered electron, is also produced when the specimen is irradiated with the primary electron beam. Together backscattered and secondary electrons contribute to the signal that reaches the scintillator and form what we refer to as the secondary electron image.

The spatial resolution of the SEM depends on the size of the electron spot which in turn depends on the magnetic electron-optical system which produces the scanning beam. The resolution is also limited by the size of the interaction volume, or the extent to which the material interacts with the electron beam. The spot size and the interaction volume are both very large compared to the distances between atoms, so the resolution of the SEM is not high enough to image down to the atomic scale, as is possible in the transmission electron microscope (TEM). The SEM has compensating advantages, though, including the ability to image a comparatively large area of the specimen; the ability to image bulk materials (not just thin films or foils); and the variety of analytical modes available for measuring the composition and nature of the specimen. Depending on the instrument, the resolution can fall somewhere between less than 1 nm and 20 nm. In general, SEM images are much easier to interpret than TEM images. As mention earlier, one of the most common use of SEM for semiconductor appliaction, when used as a microscope, is for cross-sectional analysis to dertermine device dimensions for example MOSFET channel length.

2.4.3 Transmission electron microscopy

Transmission electron microscopy (TEM) is an imaging technique whereby a beam of electrons is transmitted through a specimen, then an image is formed, magnified

and directed to appear either on a fluorescent screen or layer of photographic film, or to be detected by a sensor such as a CCD camera. TEM is a very powerful and versatile instrument that is capable of determining the structure of materials. Similar to SEM, a typical TEM comprises an illumination system, an objective lens system, a magnification system, a data recording system, a chemical analysis system and a vacuum system [72].

The main purpose of the illumination system is to extract electrons from the source and transfer them into the parallel or convergent beam on the specimen. A parallel beam is used for imaging or diffraction, while a convergent beam is used for STEM imaging, microanalysis and microdiffraction. The illumination system comprises the electron gun and the condenser lenses.

The electron gun is the heart of the TEM, which usually uses a LaB₆ thermionic emission source or a field emission source. The LaB₆ gun gives a higher illumination current but lower current density and beam coherence than that of a field emission source. A high energy electron beam in the range from 100-400 keV is typically employed, and the specimen must be specially prepared to be electron transparent (~10-1000 Å).

The condenser lenses (CL) are used to demagnify the diameter of the electron beam from its crossover point. Two condenser lenses are used in a typical TEM. CL1 forms a demagnified image of the beam and controls the spot size of the beam; CL2 converges the beam at the specimen and controls the diameter of the illuminated area of the specimen. The condenser aperture controls the fraction of the total beam flux on the sample so that it can control the illumination intensity.

The electron beam in the TEM is oriented normal to the plane of the sample, penetrates the sample, and projects an image or diffraction pattern on the viewing screen.

The objective lens (OL) is the most important lens in the system because it determines the resolution of the TEM. The OL forms an inverted and magnified image of the specimen in the first image plane and a diffraction pattern in the back focal plane. The objective aperture, inserted into the back focal plane of the objective lens, is used to control the contrast of the final image.

The electron beam is diffracted by lattice planes to form diffracted beams along different directions. Then the objective lens brings the waves traveling in the same direction (at angle θ to the optic axis) to a point on the back focal plane, forming a diffraction pattern

Fig 2.15: Schematic of diffraction pattern, projecting the diffraction pattern on the viewing screen.

A TEM image is formed by interference of Bragg scattered beams, and this interference pattern is affected by the phase modulation introduced by the aberration of

the objective lens. By focusing the intermediate lens (and hence subsequent lenses) on the back focal plane of the objective lens, a diffraction pattern may be formed on the viewing screen. This pattern contains valuable information about the crystallographic structure of the specimen. A schematic diagram of a diffraction pattern is presented in Fig. 2.15.

The capabilities of the TEM can be further extended by additional stages and detectors, sometimes incorporated on the same microscope. An electron cryomicroscope is a TEM with a specimen holder capable of maintaining the specimen at liquid nitrogen or liquid helium temperatures. This allows imaging specimens prepared in vitreous ice, the preferred preparation technique for imaging individual molecules or macromolecular assemblies. A TEM can be modified into a scanning transmission electron microscope (STEM) by the addition of a system that rasters the beam across the sample to form the image, combined with suitable detectors. An analytical TEM is one equipped with detectors that can determine the elemental composition of the specimen by analysing its X-ray spectrum or the energy-loss spectrum of the transmitted electrons

The main strength of TEM lies in its extremely high resolution, approaching 0.15 nm. However, there are a number of drawbacks to the TEM technique. Many materials require extensive sample preparation to produce a sample thin enough to be electron transparent, which makes TEM analysis a relatively time consuming process with a low throughput of samples. The structure of the sample may also be changed during the preparation process. Also the field of view is relatively small, raising the possibility that the region analysed may not be characteristic of the whole sample. There is potential that the sample may be damaged by the electron beam, particularly in the case of biological materials.

2.5 Electron mobility of MOSFET's with ultrathin gate oxide

Beside the electrical parameter, L_{eff} , the inversion-layer mobility or the effective mobility (μ_{eff}) in the channel region of a MOS transistor is considered to be another important factor, which determines the drive current and the switching speed of a MOS transistor [73]. Effective mobility (μ_{eff}) is traditionally defined in terms of the measured drain current, I_d , of a typical MOS transistor at low drain channel voltage, V_d^* , in the linear region to be [74,75]

$$I_d = \left(\frac{W_{eff}}{L_{eff}} \right) m_{eff} Q_{inv} V_d^* \quad (2.17)$$

where W_{eff} refers to the effective channel width and L_{eff} refers to the effective channel length of the transistor under test. At sufficiently large voltages above the channel threshold voltage, V_t , the inversion layer charge approaches

$$Q_{inv} \rightarrow C_{ox} (V_{gs} - V_T) \quad (2.18)$$

where V_{gs} is the gate voltage measured with respect to the source voltage which is biased at 0 V in this case. C_{ox} is the oxide capacitance per unit area, which is the value of oxide dielectric constant (ϵ_{ox}) over equivalent oxide thickness (EOT) (T_{ox}). From eq. (2.17), μ_{eff} can be further expressed as

$$m_{eff} = \frac{1}{\left(\frac{W_{eff}}{L_{eff}} \right) Q_{inv} R_{ch}} \quad (2.19)$$

where $R_{ch} = V_d^*/I_d$. V_d^* is the effective voltage across the channel of the transistor [74-77].

The effective mobility in the inversion layer follows a universal curve at room

temperature when plotted as a function of effective transverse electric field, E_{eff} , expressed as

$$E_{eff} = \frac{1}{e_{si}} (Q_{dpl} + hQ_{inv}) \quad (2.20)$$

where Q_{inv} and Q_{dpl} are the inversion and depletion carrier densities, respectively, and e_{si} is the permittivity of silicon. Here, h is an empirical parameter in defining E_{eff} in order to provide the universal relationship, where best values are $h=1/2$ for (100) electron and $h=1/3$ for holes ; (111) and (110) for electrons [78].

A typical set of data on mobility versus effective normal field for nMOSFETs is shown in Fig. 2.16 [79].

Fig. 2.16: Electron mobility in inversion layer at 300K and 77K versus effective electric field, $E_{eff} = q(N_{dpl} + h N_s) / e_{si}$ with h of $1/2$. [79].

There are three types of Coulomb scattering centers that can affect the inversion layer mobility: substrate impurities, interface state charges and charges trapped in SiO_2 . According to the Mathiessen's rule, the net mobility in MOSFET channels is controlled by three dominating mechanisms as

$$\frac{1}{\mu} = \frac{1}{\mu_{ph}} + \frac{1}{\mu_{sr}} + \frac{1}{\mu_c} \quad (2.21)$$

where μ_{ph} (phonon scattering) is the mobility controlled by phonon scattering, μ_{sr} (surface-roughness scattering) is the mobility controlled by surface roughness and μ_c (Coulomb scattering) is the mobility controlled by channel doping impurities. For this equation, the lowest mobility dominates [80,81].

Fig 2.17: Schematic diagram showing the E_{eff} or N_s dependence of mobility in inversion layer by three dominant scattering mechanisms [79].

As shown in Fig. 2.17, μ_c , μ_{ph} and μ_{sr} dominate at low, moderate and high E_{eff} regions, respectively. μ_{ph} is proportional to $E_{eff}^{-0.3}$ and only μ_{ph} depends strongly on temperature. At low temperatures, μ_{ph} is large because of weak phonon scattering such that μ_{eff} is dominated by μ_{sr} and μ_c . On the contrary, the region of E_{eff} where μ_{eff} is dominated by μ_{ph} becomes wider at higher temperature.

In recent years, as MOS transistor is being aggressively scaled down, it requires increasingly thinner gate oxide layer. It has been found that, in gate oxides thinner than 30 Å, the mobility decrease in low electric field, E_{eff} , region with decreasing oxide thickness. Several scattering mechanisms inherent to ultra thin gate oxides, such as poly-gate/oxide interface roughness scattering [82,83] and remote impurity scattering have been reported [84-88]. In particular, Remote Charge Scattering or Remote Coulomb Scattering (RCS) is extensively studied in the semiconductor industry, resulting in better theory and model [89-92]. It refers to the scattering of mobile charges in the inversion layer by charged impurities present in the gate electrode of a MOSFET. This induces the decrease in mobility with an increase in poly-Si-gate doping level and this mobility component, μ_{rcs} , has a strong E_{eff} dependence as shown in Fig. 2.18.

Although, the measurement of effective mobility is not required in the extraction of L_{eff} , however, the importance on the understanding of the effective mobility is crucial in L_{eff} extraction as MOS transistors are further scaled down. The accurate measurement of MOS transistor channel lengths becomes increasingly difficult in the deep submicron region due to the fact that the surface mobility is no longer a constant but a function of the gate voltage. As mentioned in Chapter One, due to the halo implant, the short channel transistor features a degraded effective mobility compared to the long channel reference transistor. This is because the effective channel doping increases when the halo regions

overlap as channel length gets shorter. This results in a lower effective mobility for MOS transistors with a relatively smaller metallurgical channel length, thus L_{eff} extraction approaches based on the I_V characteristics at low drain bias (or linear region) tend to overestimate the value of L_{eff} . With this basic understanding of mobility, we shall move forward to discuss in the next chapter, a new insight on Remote Charge Scattering achieved during the course of this PhD project. With this new insight, a new method of L_{eff} measurement was formulated.

Fig. 2.18: Inversion-layer electron mobility as a function of E_{eff} with T_{ox} of 3.5–1.5 nm. The solid line denotes the universal mobility curve [88].

2.6 Chapter summary

In this chapter, the theory and limitations of various L_{eff} extraction techniques are reviewed. Two widely used families of techniques are the *channel resistance based* methods and the *capacitance based* methods. The basic assumption of channel resistance based methods is that the mobility is independent of the gate length. A brief discussion regarding the validity of this assumption for state-of-the-art MOS transistor has been given in this chapter. The capacitance based methods have the advantage that they do not depend on this assumption. *SCM* based method is basically destructive and thus not suitable for routine measurement in a manufacturing environment. In addition, a brief discussion on effective mobility was presented to highlight the effect of effective mobility in L_{eff} extraction.

CHAPTER THREE

Theory of a New L_{eff} Extraction Method

The first section in this chapter gives an overview of the limitation of existing *channel resistance* L_{eff} extraction methods, which are popular and widely used in the industry. Theoretical explanation of the present limitation was presented follow by our new theory on the transistor's property when operating at low gate voltage region. We predict that the carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is only slightly above the threshold due to *remote charge scattering* (RCS). The following section then touches briefly on the device fabrication and experimental setup. The last three sections present three new L_{eff} extraction methodology base on our theory to overcome the physical limitation of extracting L_{eff} respectively.

3.1 Introduction

In almost all the conventional channel resistance based method [25-31], it was assumed that the mobility of the short channel transistor is equal to that of the long channel transistor. For MOSFETs with halo implant, the short channel transistor is actually quite different from the long channel transistor in terms of 2-D dopant distribution. According to van Meer *et al.* [23], the short channel transistor is like a transistor with significantly higher channel doping concentration compared to the long channel transistor because of the halo implant. To validate that the short channel transistor is indeed having a higher channel doping concentration compared to the long channel transistor in this work of study, a one dimension channel doping profiles between the long and short channel transistors was simulated using TSUPREM [93].

A full 0.13 μ m process simulation was performed using TSUPREM-4 (trademark of Synopsys Inc.), which is an advanced 1D and 2D process simulator for developing semiconductor process technologies. In the simulation, all the process steps used for fabricating semiconductor devices are included, for e.g. masking steps, lithography, implantation, anneal and etching. The annealing temperature used in this work is 1000°C. The model-based simulations are conducted with technology-specific conditions that resemble the actual devices being studied.

A two-dimensional silicon substrate is used as the simulation volume. Y refers to the channel depth, the direction into which the ions are implanted as shown in Fig. 3.1. A one-dimensional cutline is then placed in the channel region, along which the concentrations of the ions are extracted. This line is denoted as ID Cutline @Channel in Figs. 3.2 and Fig. 3.3.

Figure 3.1: Schematic drawing of the cross section view of the transistor where the concentrations of the ions are extracted.

Fig. 3.2 shows the one dimension channel doping profiles between a long and a short channel NMOS transistor simulated using TSUPREM. The channel doping profile refers to the ion concentration profile (ion concentration in ions/cm³ versus vertical depth into substrate in μm) taken in the channel region.

Figure 3.2: One Dimensional (1-D) Channel doping profile of (a) long channel (10 μm) and (b) short channel (0.13 μm) NMOS transistors.

Chapter 3: Theory of a New L_{eff} Extraction Method

As shown in Fig. 3.2 from our simulation, the channel doping concentration is about one order different in concentration between the short and long channel transistors. The channel doping profile simulation for PMOS is as shown in Fig. 3.3. Similarly, the doping concentration is much higher in a short channel transistor than that of a long channel transistor. This is because for short channel MOS transistors, the halo implant from the source side overlaps with the halo implant from the drain side at the channel region.

In this way, it is expected that the carrier mobility in the short channel transistor is significantly smaller than that in the long channel transistor. This is the reason why the effective channel length can be significantly over-estimated by the conventional *channel resistance* method.

Figure 3.3: One-dimensional (1-D) channel doping profile of (a) long channel (10 μm) and (b) short channel (0.13 μm) PMOS transistors.

For other methods base on the linear I-V characteristic to extract the L_{eff} , for example, the Shift-and-Ratio method [29], actually the same assumption that the mobility is a constant as a function of changing gate length was used. Hence, if those methods are applied to state-of-the-art MOS transistors with halo implants, those methods will also run into similar problems. The following 2 plots demonstrated the over-estimation of L_{eff} extracted by conventional *channel resistance* method on NMOS transistors of the 0.13 μm technology.

Fig. 3.4: L_{eff} extracted using channel resistance method by Chern et al. [26] for NMOS transistors with channel width of 20 μm of 0.13 μm technology.

The common intersection of the lines determines both DL and R_{sd} . The DL and R_{sd} extracted for the above set of MOS transistors are $-0.029 \mu\text{m}$ and 0.078 ohms respectively.

Fig. 3.5: L_{eff} extracted using channel resistance method [26] for NMOS transistors with channel width of 10 μm of 0.13 μm technology.

The DL and R_{sd} extracted for the above set of MOS transistors are $-0.018 \mu\text{m}$ and -0.79 ohms respectively. As we can see from both sets of MOS transistor under study, channel resistance method tends to yield L_{eff} greater than L_{drawn} , which is unreasonable. This problem become more serious as MOS transistor is scaled down into the sub-micron region. This can be attributed to the non-linear scaling of R_{tot} as proposed by Esseni *et al.* [59]. However, this undesirable effect from the pocket implant as devices are being scaled down may not be as strong at low V_{gs} region, as proposed in the following section.

3.2 Theory of our new approach

In state-of-the-art MOS transistors, the gate oxide thickness is below 20 Å. Krishnan et al. showed that for MOS transistors with 9-13 Å gate oxide, remote charge scattering can be observed [84]. The electron mobility in the channel becomes sensitive to the doping level of the polysilicon gate. Since the doping level of the polysilicon gate is usually much higher than the substrate, the carrier mobility in the polysilicon gate is much lower than the carrier mobility in the channel. We propose that when the gate oxide thickness approaches zero the channel mobility will approach the carrier mobility in the polysilicon gate. Take n-channel MOSFET as an example. As shown in Fig. 3.6(a) and 3.6(b), the n^+ -poly-Si is much more heavily doped than the p-well. When the gate oxide is thick, the channel mobility does not have any dependence on the n^+ -poly-Si doping concentration; however, the channel mobility depends on the channel doping concentration and decreases with the increase of the channel doping concentration. Since the channel doping concentration is much lower than the n^+ -poly doping concentration, the channel mobility is much higher than the mobility in the n^+ -poly. Therefore, if the gate oxide becomes very thin, especially when the gate voltage is small, there is strong remote charge scattering between the channel and the n^+ -poly-Si such that the channel mobility becomes smaller. When the gate oxide thickness approaches zero, the channel mobility will at most approach the mobility in the n^+ -poly-Si, as shown in Fig. 3.7(a) and 3.7(b).

Chapter 3: Theory of a New L_{eff} Extraction Method

Fig.3.6: Schematic diagram of the channel region in a n -channel MOSFET with relatively thick gate oxide (a) Low channel doping concentration; (b) High channel doping concentration.

Fig.3.7: Schematic diagram of the channel region in a n -channel MOSFET with ultra-thin gate oxide (a) Low channel doping concentration; (b) High channel doping concentration.

Chapter 3: Theory of a New L_{eff} Extraction Method

For the case of small gate voltage, the doping concentration in the p-well will not have a strong effect on the channel mobility. Fig. 3.8 shows the mobility versus gate oxide thickness plot modified from Fig. 10 of [84]. From Fig. 3.8, it was shown that the electron mobility approach a limiting value of about $50 \text{ cm}^2/\text{Vs}$ when the gate oxide thickness is scaled down. Since the inversion charge is proportional to $V_{gs} - V_{th}$, we expect that the electron mobility will approach the same limiting value of about $50 \text{ cm}^2/\text{Vs}$ at low values of $V_{gs} - V_{th}$ independent of the channel doping level difference in the long and short channel transistors due to the pocket implant. The limiting value of about $50 \text{ cm}^2/\text{Vs}$ was consistence with our own measurement which will be shown in Chapter Four. From the work of Queirolo et al. [94], the electrical and structural properties of polysilicon films deposited by low-pressure chemical vapor deposition (LPCVD) at different temperatures, doped with phosphorus by ion implantation and isochronally annealed at temperatures ranging from 600°C to 1000°C were characterized. With our device having similar process conditions and properties, it was estimated that the mobility of a heavily doped n-type polysilicon is in the range of 30 to $40 \text{ cm}^2/\text{Vs}$ as reported by Queirolo et al.; we believe that the limiting value of $50 \text{ cm}^2/\text{Vs}$ originates from the electron mobility of the polysilicon gate.

Electron mobility with gate oxide scaling.

Fig.3.8: Mobility versus gate oxide thickness plot modified from Fig.10. of [84]. The measured electron mobility is seen to decrease as the inversion charge density of the channel decrease for a given T_{ox} .

To better understand remote charge scattering, Fig. 3.9 (c) shows that the bottom of the conduction band of the polysilicon gate is significantly below the bottom of the conduction band of the channel in energy when $V_{gs} - V_{th}$ is large. This results in weak quantum mechanical tunneling interaction between the conduction band of the polysilicon gate and the conduction band of the channel. When $V_{gs} - V_{th}$ is small as shown in Fig 3.9(b), the bottom of the conduction band of the polysilicon gate is only slightly below the bottom of the conduction band of the channel in energy, resulting in relatively strong quantum mechanical tunneling interaction between the conduction band of the polysilicon gate and the conduction band of the channel. With very thin gate oxide thickness, remote charge scattering is particularly strong when $V_{gs} - V_{th}$ is small thus the electron mobility

Chapter 3: Theory of a New L_{eff} Extraction Method

in the channel becomes sensitive to the doping level of the polysilicon gate and is independent of the channel doping level difference in the long and short channel transistors. This is supported by theoretical calculations. [86]

(a)

(b)

Fig.3.9: Energy band diagram of a cross-section of a n-channel MOSFET. (a) at $V_{gs} = 0$; (b) V_{gs} slightly greater than V_{th} ; (c) V_{gs} strongly greater than V_{th} .

Similar energy band diagram for p-channel MOSFET was shown in Fig. 3.10. At thermal equilibrium $V_{gs} = 0$, the work function differences in p-channel MOSFET and n-channel MOSFET are symmetric as shown in Fig 3.10 (a). The same theory can be applied to the p-channel MOSFET as shown in Fig. 3.10 (b) and (c).

Fig.3.10: Energy band diagram of a cross-section of a p-channel MOSFET. (a) at $V_{gs} = 0$; (b) V_{gs} slightly greater than V_{th} ; (c) V_{gs} strongly greater than V_{th} .

3.3 Remote charge scattering at low gate voltage

We predict that the carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is only slightly above the threshold. This is because of remote Coulomb scattering (RCS) caused by the doping impurities in the gate polysilicon when the gate oxide is very thin [84-92]. Remote Coulomb scattering may not be important for gate voltage much larger than threshold but our theory predicts that it can be important at small gate voltage close to the threshold or below. In the sub-threshold region, there is no screening from electrons due to strong inversion. We propose that the electron mobility is controlled by μ_{RCS} , the RCS mobility due to the doping impurities in the n^+ gate polysilicon and μ_{CS} , the mobility due to Coulomb scattering (CS) from the doping impurities in the channel region of the MOS transistor according to the following equations:

$$1/\mu_{n,short} = 1/\mu_{RCS} + 1/\mu_{CS,short} \quad (3.1)$$

$$1/\mu_{n,long} = 1/\mu_{RCS} + 1/\mu_{CS,long} \quad (3.2)$$

$\mu_{n,short}$ and $\mu_{n,long}$ are the electron mobilities of the short transistor and long transistor respectively. $\mu_{CS,short}$ and $\mu_{CS,long}$ are the CS electron mobilities of the short transistor and long transistor respectively. In the above equations, μ_{RCS} is the same for both short and long transistors. This is because the doping concentration in the polysilicon gate is the same for short and long transistors. $\mu_{CS,short} < \mu_{CS,long}$ because the channel doping concentration is higher for the short transistor than the long transistor. As shown in Fig. 3.11(a) when the gate voltage $V_{gs} \gg V_{th}$, there is a strong screening effect from electrons due to inversion such that μ_{RCS} , $\mu_{CS,short}$ and $\mu_{CS,long}$ increases with $V_{gs} - V_{th}$. Our theory is that the screening effect is much stronger for RCS than for CS such that $\mu_{RCS} \gg \mu_{CS,long} > \mu_{CS,short}$. This is expected because for RCS the scattering centers and the

Chapter 3: Theory of a New L_{eff} Extraction Method

channel electrons are physically more separated than the case for CS. Hence the effect of RCS is not strong for $V_{gs} \gg V_{th}$ and this is consistent with the work of Romanjek et al.[95]. According to Fig.3 of [95], the electron mobility measured by their new method is approximately equal for the short and long channel transistors with pocket implant when the inversion charge is small at small gate voltage. Thus our theory is consistent with the latest experimental data. In addition, our own experimental data presented in section 4.2.1 matches well with our theory. When the gate voltage $V_G < V_{th}$ for the sub-threshold region, there is basically no strong screening effect such that μ_{RCS} , $\mu_{CS,short}$ and $\mu_{CS,long}$ depends very much on the gate polysilicon doping concentration, the channel doping concentration in the short transistor and the channel doping concentration in the long transistor. However, the channel doping concentration is usually $< 10^{18} \text{ cm}^{-3}$ for both the short and long transistors according to our simulations while the n^+ gate polysilicon doping is about 10^{20} cm^{-3} . Then it is expected that in the sub-threshold regime, $\mu_{RCS} < \mu_{CS,short} < \mu_{CS,long}$ such that $\mu_{n,short} = \mu_{n,long} = \mu_{RCS}$ approximately as shown in Fig.3.11(b).

Figure 3.11: (a) RCS and CS components of mobility close to threshold, (b) Mobility of short and long MOS transistors close to threshold.

3.4 Device fabrication and experimental setup

The MOS transistors under study were fabricated using 0.13 μm , 0.11 μm and 0.1 μm CMOS technology on 200 mm p -type Si substrate. For these generations of CMOS technology, the gate oxide was grown by rapid thermal oxidation with thickness smaller than 20 Å, with S/D extension structures, halo/pocket implant and L-shape spacer. Cobalt salicide was used for lowering resistance and shallow trench isolation (STI) was used for device isolation.

Cross-sectional transmission electron microscopy (XTEM) was used to measure the physical gate length L_{gate} . We found that L_{gate} is very slightly smaller than the drawn gate length L_{drawn} . In addition, Scanning Capacitance Microscopy (SCM) was used to do direct channel length measurement [68-70].

All current-voltage (I-V) measurements involved in our method were measured using an HP4156B Precision Semiconductor Parametric Analyzer.

Unless otherwise specified, the reference MOS transistor used in this research work were long channel device with L_{mask} of 10 μm and channel width of 10 μm .

Chapter 3: Theory of a New L_{eff} Extraction Method

One point to mention in this section is the proper layout of test structure. The major objective of our new method is to reduce the sensitivity of mobility variation with respect to channel length by using the remote charge scattering limit. However, the mobility is also affected by stress effect due to shallow trench isolation, (STI). Mechanical stress from STI was found to cause up to 19% variation in 0.18 μm technology SOI device [96] and with the scaling down of CMOS technology, MOSFET characteristics become more sensitive to the device layout pattern [97]. The compressive stress from STI edge affects many device characteristics, for example, carrier mobility, threshold voltage (V_{th}) and transconductance (g_m).

In transistor design layout, the active-area length (L_a : length from the gate edge to the STI edge) has significant influence on device characteristics even though the transistors' width and length remain unchanged. Both V_{th} and g_m are affected by active-area layout in n-channel MOSFET and p-channel MOSFET. Furthermore, the influence of layout pattern is different between n-channel MOSFET and p-channel MOSFET. Device characteristics are dependent on the active-area length, because of the large compressive stress, diffusion of pocket impurity (boron) is reduced, and V_{th} becomes higher in n-channel MOSFET. Also, it increases hole mobility, and thus, I_{dsat} , in p-channel MOSFET. Therefore, proper test structure layout is also essential in accurate L_{eff} extraction by minimizing the variation in carrier mobility. In addition, optimization of the process in reducing the stress is also crucial [98,99].

3.5 New methodology

3.5.1 Methodology I: A Modified Shift-and-Ratio Method

For this Modified Shift-and-Ratio method, the measurement is performed at a gate voltage range close to the threshold voltage such that the mobility of the short channel transistor with halo implant is approximately equal to the mobility of the long channel transistor with the same halo implant. However, it was found that the shift-and-ratio method is very difficult to use at a gate voltage range close to the threshold voltage because of the shift. Hence, we decided to throw away the shift in the original shift-and-ratio method. Instead, we directly measured the threshold voltage difference between the short channel transistor and the long channel transistor. This is the key point of our proposed method. It turns out that the algorithm of our method is also much simpler than the original shift-and-ratio method such that it can be easily implemented into automatic wafer level electrical testing systems.

However, since our method makes use of the direct measurement of the threshold voltage, the choice of the V_{th} extraction method can affect the resulting L_{eff} extracted by our modified Shift-and-Ratio method. The constant-current (CC) method was used in this case because of its simplicity and the consistence of the L_{eff} extracted. The threshold voltage is defined as the V_{gs} when $I_{ds} = 0.1 * (W_{eff}/L_{eff}) \mu A$.

The drain current in the linear region of MOS transistors is given by

$$I_{ds} = \mu(V_{gs})C_{ox}(W_{eff}/L_{eff})(V_{gs} - V_{th} - 0.5V_{ds})V_{ds} \quad (3.3)$$

where I_{ds} and V_{ds} are the source-drain current and voltage respectively, V_{gs} is the gate-source voltage, V_{th} is the linear threshold voltage, C_{ox} is the gate oxide capacitance per unit area, W_{eff} is the effective gate width, $\mu(V_{gs})$ is the mobility and L_{eff} is the effective channel length.

Chapter 3: Theory of a New L_{eff} Extraction Method

In the original shift-and-ratio method, the total drain-source resistance R_{tot} is modified to include the parasitic source/drain resistance, R_{sd} :

$$R_{tot} = V_{ds} / I_{ds} = R_{sd} + \left[L_{eff} / \left(\mu(V_{gs}) C_{ox} W_{eff} (V_{gs} - V_{th} - 0.5V_{ds}) \right) \right] \quad (3.4)$$

We can define $S(V_{gs})$ as the derivative of R_{tot} with respect to V_{gs} .

$$S(V_{gs}) = \partial R_{tot} / \partial V_{gs} \quad (3.5)$$

In general, L_{eff} extraction is done by considering a short channel MOS transistor with $L_{eff} = L_i$ and a long channel MOS transistor with $L_{eff} = L_o$ which serves as the reference device. The ratio L_o/L_i can be shown to be equal to the ratio $r(V_{gs})$ of the derivative $S_o(V_{gs})$ for the reference MOS transistor and the derivative $S_i(V_{gs} + \Delta V_{th})$ for the MOS transistor under test with the same gate width but shorter channel length as follows:

$$r(V_{gs}) = S_o(V_{gs}) / S_i(V_{gs} + \Delta V_{th}) = (L_o / \mu_o) / (L_i / \mu_i) \quad (3.6)$$

$$\text{where } \Delta V_{th} = V_{tho} - V_{thi}$$

$$L_{eff} = L_i = L_{drawn} - \Delta L \quad (3.7)$$

Subscripts o and i refer to the reference device (long transistor) and the specific device under test (short transistor) respectively. Note that R_{sd} is assumed to be constant with gate bias, and so it can be conveniently removed in the derivatives. In the original shift-and-ratio method, it is assumed that the mobility is a function of $(V_{gs} - V_{th})$ but the mobility is not a function of the channel length such that $\mu(V_{gs} - V_{th})$ is common for the two devices. However, in our modified method, we proposed that the long channel transistor mobility and the short channel transistor mobility are not equal. For the original shift-and-ratio method, in order to solve L_{eff} , ΔV_{th} is adjusted until $r(V_{gs})$ becomes a constant function of V_{gs} , once a constant ratio of $r(V_{gs})$ is achieved, L_{eff} is calculated using Eq. (3.7) with the assumption that $\mu_o = \mu_i$.

Chapter 3: Theory of a New L_{eff} Extraction Method

However, it can be difficult to determine $r(V_{gs})$ as a constant function of V_{gs} by evaluating the mean square deviation of $r(V_{gs})$ as a function of ΔV_{th} as proposed by Taur et al.[29]. When the spread of the derivative is large as shown in Fig. 3.12, it will be difficult to determine the sharp minimum point that are required to find the best match (constant ratio). Moreover, it is more complicated to implement the algorithm in automatic wafer electrical testing systems for large volume of wafer testing. There is one more serious problem; the mobility of the long transistor and the mobility of the short transistor are different, especially for transistors with halo implants. By experience, we found that it is rather difficult to perform the shift for a gate voltage range close to the threshold voltage.

Fig.3.12: Average ratio (open dot) and variance (solid dot) versus gate voltage shift. Variance spread over some values, difficult to determine the correct minimum point for constant ratio.

Chapter 3: Theory of a New L_{eff} Extraction Method

For our proposed modification of the shift-and-ratio method, $r(V_{gs})$ is determined by direct measurement of the threshold voltage, V_{th0} and V_{thi} , of the reference device and specific device under test. Instead of trying to find the best match, V_{th0} and V_{thi} can be easily measured using existing constant-current extraction method that is widely used in the industry [100]. All the threshold voltages used in the experiment were extracted at a constant current of $(W/L)0.1\mu A$, where W is the channel width and L is the channel length. This threshold voltage extraction method is simple to use and less dependent on the R_{sd} of the MOS transistor. Then ΔV_{th} is computed and used to find $r(V_{gs})$. Once $r(V_{gs})$ is found, L_{eff} is calculated using Eq. (3.7) because L_o is considered known for the long channel reference device. This method is easier to implement because there is no need to find the minimum point. In case the minimum point cannot be found, this method will still yield a L_{eff} . This method can be easily integrated into automatic wafer electrical test systems with virtually no additional measurement data needed. The most important point is that this method can be applied to a gate voltage range close to the threshold voltage where we hypothesized that the short and long channel transistors with the same halo implant will have approximately equal carrier mobilities. However, since $r(V_{gs})$ was not found by adjusting ΔV_{th} until it becomes a constant function of V_{gs} for our method compared to that of the original Shift-and-Ratio method, our method will not be robust enough in minimizing the effect of parasitic source-drain resistance. In the case of the original Shift-and-Ratio method, averaging out $r(V_{gs})$ to become a constant function of V_{gs} helps to eliminate the effect of parasitic source-drain resistance.

The V_{gs} ranges used to extract the L_{eff} was from 0.1 V to 0.5 V; within this range our MS&R method can still be applicable to extract L_{eff} . As shown in Eq. (3.3), the drain

current in the linear region and slightly below threshold voltage region of MOS transistors, can be generalized to the form;

$$I_{ds} = m(V_{gs})C_{ox}(W_{eff}/L_{eff})f(V_{gs} - V_{th})V_{ds} \quad (3.8)$$

Subsequently the total drain-source resistance R_{tot} is modified to include the parasitic source/drain resistance, R_{sd} :

$$R_{tot} = V_{ds}/I_{ds} = R_{sd} + [L_{eff}/(m(V_{gs})C_{ox}W_{eff}f(V_{gs} - V_{th}))] \quad (3.9)$$

This extends the validity of the equation from the linear region to slightly below threshold voltage region of the $I_{ds} - V_{gs}$ curve.

3.5.2 Methodology II: A Modified Shift-and-Ratio Method for extraction of relative change in L_{eff}

As discussed in earlier section, when pocket implant is incorporated in deep submicron MOS transistors to suppress punchthrough and to improve the threshold voltage V_{th} roll-off. The physical properties between a long channel transistor and a short channel transistor can be greatly different [23]. The effective channel-doping, N_{eff} increase if the halo regions overlap in the short channel transistor.

To eliminate the big difference in the physical properties between the long and short channel transistors, a short channel transistor with nearly the same channel length is used as the reference transistor instead of a long channel transistor when extracting the L_{eff} . Here we are not extracting the absolute L_{eff} values but the relative changes in L_{eff} compared to a reference transistor. The L_{eff} values of the reference transistor can be extracted in the first place by using other methods, for example, capacitance based method [16, 41-48], Modified-Shift-and-Ratio (MS&R) L_{eff} extraction method [29] or Scanning Capacitance Microscopy (SCM) [68-70], etc.

Chapter 3: Theory of a New L_{eff} Extraction Method

Devices were fabricated using the standard 0.13 μm CMOS technology on 200 mm p -type Si substrate as mention in Section 3.4. To further evaluate, the capability of this new idea, four different poly etch trimming process was performed on four different wafer as follow: wafer 1, 0% over-etch trim; wafer 2, 25% over-etch trim; wafer 3, 50% over-etch trim; wafer 4, 75% over-etch trim. Modified Shift-and-Ratio method was applied to the short channel transistor with channel length of 0.13 μm and channel width of 20 μm on all four wafers. A total of nine sites on each wafer was measurement with site number 6 of wafer one chosen to be the reference transistor for all the calculation of the relatively change in L_{eff} , the result will be discussed in Chapter 4.

3.5.3 Methodology III: A Subthreshold Regime-based Method

In this section, we present a novel method for accurate L_{eff} extraction based on the sub-threshold regime. We propose that the electron mobility is approximately equal for short and long channel MOS transistors in the sub-threshold region when the gate oxide is very thin and it is possible to perform L_{eff} measurement in the sub-threshold region of MOS transistors with ultra-thin gate oxide. The methodology of our new L_{eff} extraction algorithm will be presented. In the sub-threshold regime, the impact from channel mobility variation with gate length is minimized. Furthermore, the measurement is independent of parasitic series resistance and extraction of threshold voltage is not required. Fig. 3.11 shows an example of sub-threshold swing (S) plotted versus L_{drawn} .

Fig.3.13: Example of sub-threshold swing (S) plotted versus L_{drawn} for NMOS transistor.

It shows an exponential dependence of S on L_{drawn} .

Fig.3.14: Drift and diffusion components of current in an I_{ds} - V_g plot. Their sum is a total current represented by the solid curve [101].

Fig. 3.14 shows the drift and diffusion components of drain current. In the region with gate voltage above threshold voltage (strong inversion region), the drift current dominates. In the subthreshold region, conduction is dominated by the diffusion current. Unlike the drift current which is proportional to the electric field, the diffusion current is proportional to the spatial derivative of the electron density. In the weak inversion where $y_B < y_S < 2y$, the diffusion current dominates.

For this new method, the L_{eff} extraction is conducted within the regime of the MOSFET's subthreshold region such that the drain current can be expressed as [101]

$$I_{ds} = m_{eff} C_{ox} \frac{W_{eff}}{L_{eff}} (m-1) \left(\frac{kT}{q} \right)^2 e^{q(V_{gs}-V_{th})/mkT} (1 - e^{-qV_{ds}/kT}) \quad (3.10)$$

where I_{ds} and V_{ds} are the drain-source current and voltage respectively, V_{gs} is the gate-source voltage, V_{th} is the threshold voltage, C_{ox} is the gate oxide capacitance per unit area,

Chapter 3: Theory of a New L_{eff} Extraction Method

W_{eff} is the effective gate width, m_{eff} is the effective mobility which is approximately the same between long channel and short channel transistor at V_{gs} in the sub-threshold region and L_{eff} is the effective channel length. m is the *body-effect coefficient* and it can be expressed as shown below,

$$m = 1 + \frac{\sqrt{\epsilon_{si} q N_a / 4 \gamma_B}}{C_{ox}} \quad (3.11)$$

where ϵ_{si} is the silicon permittivity, q is the electronic charge, N_a is the acceptor doping concentration, γ_B is the difference between the Fermi level in the channel region and the intrinsic level. m is related to the *sub-threshold swing*, S by

$$S = 2.3 \frac{mkT}{q} \quad (3.12)$$

Equation (3.10) can be simplified into the form as shown below

$$I_{ds} = a e^{b(V_{gs} - V_{th})} \quad (3.13)$$

where

$$a = m_{eff} C_{ox} \frac{W_{eff}}{L_{eff}} (m-1) \left(\frac{kT}{q} \right)^2 (1 - e^{-qV_{ds}/kT}) \quad (3.14)$$

$$\text{and } b = q/mkT = 2.3/S \quad (3.15)$$

To extract L_{eff} , (3.14) can be further simplified as shown below

$$a = \frac{a_o (m-1)}{L_{eff}} \quad (3.16)$$

$$\text{where } a_o = m_{eff} C_{ox} W_{eff} \left(\frac{kT}{q} \right)^2 (1 - e^{-qV_{ds}/kT})$$

Rearranging (3.16), we have:

Chapter 3: Theory of a New L_{eff} Extraction Method

$$\mathbf{a}_0 = \frac{\mathbf{a}L_{eff}}{(m-1)} \quad (3.17)$$

It is assumed that $\mathbf{m}_{eff}$ is approximately equal in the sub-threshold region for both short channel and long channel transistors, C_{ox} and W_{eff} are equal for both short channel and long channel transistor with the same basing V_{ds} and temperature. Therefore α_0 can be assumed to be constant for both the short and long channel transistors as follow,

$$\mathbf{a}_0 = \frac{\mathbf{a}_l L_l}{(m_l - 1)} = \frac{\mathbf{a}_s L_s}{(m_s - 1)} \quad (3.18)$$

where L is the channel length, m is the *body-effect coefficient* and α is defined in (3.14)

Subscripts l and s refer to the reference device and the specific device under test respectively.

As such, L_{eff} can be extracted using (3.18),

$$L_{eff} = L_s = \frac{\mathbf{a}_l L_l (m_s - 1)}{\mathbf{a}_s (m_l - 1)} \quad (3.19)$$

The important step proposed to extract L_{eff} , is to determine the parameter α and m for both the short and long channel transistor. L_l is the channel length of a long channel transistor taken to be 10 μm .

To determine α and m , the sub-threshold swing S was extracted to determine m and $\mathbf{b}$ using (3.12) and (3.15) respectively; the temperature in (3.12) and (3.15) is room temperature. The next step is to determine V_{th} . We have tested many methods, including the constant current (CC) method [102], maximum transconductance (MAX GM) method [92], transconductance change (TC) method [103], match point (MP) method [104], second derivative of the logarithm of drain current (SDL) method [105], transconductance linear extrapolation (GMLE) method [106], maximum G-function (MAX G-FUNCTION)

Chapter 3: Theory of a New L_{eff} Extraction Method

method [107], normalized mutual integral difference (NMID) method [108] and the “Third Derivative of Drain-Source Current” (TD) method [109]. (The detailed theory of the various methods to extract threshold voltage will be discussed in the Appendix B.) We found that those methods involving differentiation or extrapolation or an arbitrary number are not suitable. We found that the NMID method, which is based on integration and thus not sensitive to measurement noise, gives the most consistent results for L_{eff} determination. One point to note here is that V_{th} was extracted in the linear region and it is not really related to the diffusion current components as defined in the subthreshold current equation. However, the difference between both the extracted V_{th} can be negligible. In this part of the study, various methods of extracting V_{th} as mentioned above were performed and the resulting L_{eff} extracted was evaluated. The results will be presented in Chapter 5 of the thesis. With V_{th} extracted using the NMID method [108], a can be easily found by performing a nonlinear least squares fitting of the subthreshold current using (3.13). With the extracted values of b and V_{th} fixed, α was varied to determine the best fit of the I-V sub-threshold curve using (3.13). The region of the I-V sub-threshold curve used was chosen to be below the V_{th} , and V_{gs} range used was within 0.1V which was close to the region where the subthreshold slope was extracted. As shown in Fig. 3.15, for a long channel transistor used as reference in this example, the value for a is equal to 3.27×10^{-7} . Similarly, a and V_{th} can be determined for a short channel transistor as shown in Fig. 3.16. For the short channel transistor with L_{drawn} of $0.12 \mu\text{m}$, α is equal to 8×10^{-5} . Once α and m are determined for both the long and short channel transistors, L_{eff} can be extracted using equation (3.19).

The region of the I-V sub-threshold curve used was chosen to be below the V_{th} , and V_{gs} range use was within 0.1V, which was close to the region where the subthreshold

Chapter 3: Theory of a New L_{eff} Extraction Method

slope was extracted. For a device with a constant subthreshold slope over the subthreshold region, this method can be applied over the full range of the subthreshold region. However, if the subthreshold slope varies across the subthreshold region, it was advice to extract the L_{eff} within a smaller range below the V_{th} . For example, if there is a hump in the I-V characteristics due to non-optimization of the STI process, L_{eff} has to be extracted within a smaller range below the V_{th} . In the case of the device experiencing punchthrough, our new method will be unable to yield a reasonable L_{eff} values.

Fig. 3.15: Nonlinear least squares fitting of the subthreshold current of a long channel transistor. The value generated for **a** is equal to 3.27x10⁻⁷ and V_{th} is 0.29 V.

Fig.3.16: Nonlinear least squares fitting of the subthreshold current of a short channel transistor. The value generated for **a** is equal to 8×10^{-5} and V_{th} is 0.33 V

3.6 Chapter summary

In summary, we have proposed that the carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is only slightly above the threshold due to RCS. Based on this theory, three types of L_{eff} extraction method are developed to overcome the limit of conventional L_{eff} extraction algorithms. However, our new methods do have their limitation of not being robust enough to be applied to automatic wafer electrical test systems for routine process monitoring. Nevertheless, our results indicate that our theory of the carrier mobility is approximately equal for both the

Chapter 3: Theory of a New L_{eff} Extraction Method

short and long channel transistors when the gate voltage is only slightly above the threshold due to RCS is correct. The results and discussion will be presented in the following chapters.

CHAPTER FOUR

Results and Discussion on Modified Shift-and-Ratio Method

This chapter consists of two sections; each presents the results generated by our proposed MS&R method, tested on both n-channel Metal-Oxide-Silicon (NMOS) and p-channel Metal-Oxide-Silicon (PMOS) transistors fabricated by state-of-the-art Complementary Metal-Oxide-silicon (CMOS) technology. The extracted L_{eff} value by our MS&R method was compared with the original S&R method and also the capacitance based methods. In addition, we have studied the cross-sections of the MOSFETs using SCM and by comparing our extracted L_{eff} measurement with that of the SCM measurement, we are able to show the consistency of our L_{eff} extraction method.

4.1 Introduction

The E-test setup for our L_{eff} extraction method is as follow:

The source voltage and substrate voltage of the MOS transistors were tied to ground ($V_s=V_b=0$). The gate voltage (V_g) was changed from 0 V to 1 V in the steps of 0.001 V. The drain voltages (V_d) of the MOS transistors were set at both 0.05 V and 0.1 V. The drain current (I_{ds}) was then measured for both sets of biasing conditions.

The electrical testing data measured was used to extract the L_{eff} using the Shift-and-Ratio method and our modified method. L_{eff} was calculated for various V_{gs} ranges of 0.1 V to 0.9 V, 0.2V to 0.5 V, 0.2 V to 0.6 V and 0.1 V to 0.5 V.

For the 0.13 μm CMOS technology, the MOS transistor under study has a channel width of 10 μm and L_{mask} of 1 μm , 0.5 μm , 0.3 μm , 0.2 μm , 0.15 μm , 0.13 μm and 0.12 μm . For the 0.1 μm CMOS technology, the MOS transistor under study has a channel width of 10 μm and L_{mask} of 1 μm , 0.5 μm , 0.3 μm , 0.2 μm , 0.11 μm , 0.09 μm and 0.08 μm .

In addition, for the 0.11 μm CMOS technology, L_{eff} extraction using capacitance method was performed and compared with our extracted L_{eff} measurement. For the 90 nm CMOS technology, the MOS transistor under study has a channel width of 10 μm and L_{mask} of 0.08 μm .

4.2 Results and discussion I

This section presents the results of applying the new Modified Shift-and-Ratio method across 4 CMOS technologies.

The following four figures compare the L_{eff} extracted using Shift-and-Ratio method and our improved method over different V_{gs} ranges of 0.13 μm technology CMOS for $V_d = 0.05\text{ V}$ and $V_d = 0.1\text{ V}$. L_{eff} was calculated for various V_{gs} ranges of 0.1 V to 0.9 V, 0.2 V to 0.6 V and 0.1 V to 0.5 V.

Fig. 4.1: Extracted L_{eff} with $V_d = 0.05\text{ V}$ for various V_{gs} ranges of NMOS transistor for 0.13 μm CMOS technology.

Fig. 4.2: Extracted L_{eff} with $V_d = 0.1V$ for various V_{gs} ranges of NMOS transistor for 0.13 μm CMOS technology.

Fig. 4.3: Extracted L_{eff} with $V_d = 0.05V$ for various V_{gs} ranges of PMOS transistor for 0.13 μm CMOS technology.

Fig. 4.4: Extracted L_{eff} with $V_d = 0.1$ V for various V_{gs} ranges of PMOS transistor for 0.13 μm CMOS technology.

From Fig. 4.1 to Fig. 4.2, we can see that the Shift-and-Ratio method over-estimates the value of L_{eff} for NMOS transistor. From Fig. 4.3 to Fig. 4.4, L_{eff} was over-estimated for PMOS transistors when L_{drawn} is less than 0.3 μm . By using our Modified-Shift-and-Ratio method, a more reasonable value of L_{eff} was extracted for V_{gs} of 0.1 to 0.5 V. L_{eff} extracted is not much different between two different drain bias of $V_d = 0.05$ V and $V_d = 0.1$ V. The measured electrical test data for $V_d = 0.05$ V and V_{gs} range of 0.1 V to 0.5 V was used subsequently to extract L_{eff} of the MOS transistor for the 0.13 μm , 0.11 μm and 0.1 μm technology. $V_d = 0.05$ V was used because we want to make sure that the MOS transistors were in the linear region when testing.

This improved method is applied on MOS transistors fabricated by 0.13 μm and advanced 0.1 μm CMOS technologies. Both V_{gs} ranges of 0.1 V to 0.5 V and 0.4 V to 0.9 V were applied to extract the L_{eff} . The effective channel lengths extracted from NMOS

data (V_{gs} ranges of 0.1 V to 0.5 V) versus mask length are shown in Fig. 4.5(a) for 0.13 μm technology and in Fig. 4.5(b) for 0.1 μm technology. The effective channel lengths extracted were reasonably acceptable. $DL = L_{drawn} - L_{eff}$, is fairly constant around 0.018 μm and 0.008 μm for 0.13 μm technology and 0.1 μm technology, respectively .

Fig. 4.5a: Extracted channel length, L_{eff} , and delta L, DL , of NMOS transistors using modified method for 0.13 μm technology. (solid dot) represents the L_{eff} and (open dot) represents the DL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.018 μm .

Fig. 4.5b: Extracted channel length, L_{eff} , and delta L , DL , of NMOS transistors using modified method for 0.1 μm technology. (solid dot) represents the L_{eff} and (open dot) represents the DL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.008 μm .

The effective channel lengths extracted from PMOS data (V_{gs} ranges of 0.1 V to 0.5 V) versus mask length are shown in Fig. 4.6(a) for 0.13 μm technology and in Fig. 4.6(b) for 0.1 μm technology. The effective channel lengths extracted from the PMOS were reasonably acceptable with DL around 0.033 μm and 0.015 μm for 0.13 μm technology and 0.1 μm technology, respectively. For both technologies, we can see that the DL calculated for PMOS is about two times larger than that of the NMOS. This can be explained, as boron is normally used in source/drain implant for PMOS transistor and arsenic for NMOS transistor. Boron has a much higher diffusion coefficient than arsenic,

resulting in shorter effective channel length for the PMOS transistor compared to the NMOS transistor even though the gate lengths can be the same.

Fig. 4.6a: Extracted channel length, L_{eff} , and delta L, DL , of PMOS transistors using modified method for 0.13 μm technology. In the figure, (solid dot) represents the L_{eff} and (open dot) represents the DL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant at around 0.033 μm .

Fig. 4.6b: Extracted channel length, L_{eff} , and delta L, DL , of PMOS transistors using modified method for 0.1 μm technology. In the figure, (solid dot) represents the L_{eff} and (open dot) represents the DL . Dashed line simply represents $L_{eff} = L_{drawn}$ and is given for reference. DL is fairly constant around 0.015 μm .

The original shift-and-ratio method was also applied to the same MOS transistors under study. Table 4.1 and Table 4.2 show the effective channel lengths extracted for NMOS transistors using the shift-and-ratio method and our modified method for both generations of CMOS technology. The effective channel lengths were extracted in both V_{gs} range of 0.4 V to 0.9 V and 0.1 V to 0.5 V using both our modified and original shift-and-ratio method. For both our modified method and original shift-and-ratio methods, the effective channel lengths extracted in the V_{gs} range of 0.4 V to 0.9 V results in unrealistic high values for deep submicron transistors with high dose halo/pocket implants. This is consistent with the observation reported in [23]. For this V_{gs} range of 0.4 V to 0.9 V, both methods yielded very close L_{eff} values. This is an indication that our modified method is

comparable with that of the original shift-and-ratio method. The effective channel lengths extracted for NMOS transistors using our modified method in the V_{gs} range of 0.1 V to 0.5 V give more reasonable results however for this V_{gs} range, we are unable to obtain a value for L_{eff} using the original shift-and-ratio method. Similarly, the original shift-and-ratio method tends to overestimate the effective channel length for PMOS transistors as shown in Table 4.3 for 0.13 μm technology and Table 4.4 for 0.1 μm technology.

Comparing the extracted effective channel lengths using both shift-and-ratio method and our modified method in the V_{gs} range of 0.1 V to 0.5 V, our method yields more reasonable values but shift-and-ratio method was unable to yield L_{eff} . Comparing the extracted effective channel lengths using both shift-and-ratio method and our modified method in the V_{gs} range of 0.4 V to 0.9 V, our method yields very similar values with that of the shift-and-ratio method. For NMOS transistors, our extracted values for L_{eff} are very close to the physical gate lengths for both the 0.13 μm and 0.1 μm technology. This result is not surprising to us because there are no major processing steps different between these two technologies. The physical gate length different between the two technologies is about 0.04 μm , the smaller gate length for the 0.1 μm technology was achieved by over etching and trimming of the poly-Si gate. There is no difference in the spacer and S/D extension implant between the two CMOS technologies under study in this case. Although the threshold voltage adjustment implant and halo implant are slightly different between the two technologies; but from the extracted L_{eff} , it was observed that there is no significant different in DL between these two technologies. For PMOS transistors, our extracted values for L_{eff} are slightly smaller than the physical gate lengths. This is expected because boron is typically used in PMOS transistor for S/D implant. Compare to arsenic or phosphorus that is typically used in NMOS transistor, boron has a higher

Chapter 4: Results and Discussion on Modified Shift-and-Ratio Method

diffusivity. Therefore, the S/D region tends to diffuse further into the channel, resulting in a smaller L_{eff} value for PMOS transistor. For both NMOS and PMOS transistors fabricated by the two generations of CMOS technology, the original shift-and-ratio method tends to yield L_{eff} values much higher than the physical gate length. This is actually quite unreasonable as pointed out in [23].

Table 4.1: A comparison of extracted L_{eff} values for NMOS between our modified Shift-and-Ratio method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.13 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)		0.12	0.13	0.15	0.20	0.30	0.50
L_{gate} (μm)		0.105	0.115	0.135	0.185	0.285	0.485
L_{eff} (μm)	Modified S&R (V _{gs} range of 0.1 V to 0.5 V)	0.1040	0.1146	0.1348	0.1786	0.2833	0.4794
	Modified S&R (V _{gs} range of 0.4 V to 0.9 V)	0.1455	0.1556	0.1744	0.2371	0.3589	0.5587
	S&R (V _{gs} range of 0.4 V to 0.9 V)	0.1499	0.1531	0.1768	0.2371	0.3688	0.5815

Table 4.2: A comparison of extracted L_{eff} values for NMOS between our modified Shift-and-Ratio method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)		0.08	0.09	0.11	0.20	0.30	0.50
L_{gate} (μm)		0.075	0.085	0.105	0.195	0.295	0.495
L_{eff} (μm)	Modified S&R (V _{gs} range of 0.1 V to 0.5 V)	0.0741	0.0838	0.1024	0.1917	0.2938	0.4851
	Modified S&R (V _{gs} range of 0.4 V to 0.9 V)	0.1031	0.1137	0.1356	0.2024	0.3216	0.5542
	S&R (V _{gs} range of 0.4 V to 0.9 V)	0.1099	0.1147	0.1327	0.2027	0.3216	0.6091

Chapter 4: Results and Discussion on Modified Shift-and-Ratio Method

Table 4.3: A comparison of extracted L_{eff} values for PMOS between our modified Shift-and-Ratio method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.13 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)		0.12	0.13	0.15	0.20	0.30	0.50
L_{gate} (μm)		0.105	0.115	0.135	0.185	0.285	0.485
L_{eff} (μm)	Modified S&R (Vgs range of 0.1 V to 0.5 V)	0.0835	0.0974	0.1232	0.1628	0.2704	0.4653
	Modified S&R (Vgs range of 0.4 V to 0.9 V)	0.1161	0.1267	0.1478	0.1936	0.2960	0.4859
	S&R (Vgs range of 0.4 V to 0.9 V)	0.1160	0.1161	0.1343	0.1849	0.2643	0.4421

Table 4.4: A comparison of extracted L_{eff} values for PMOS between our modified Shift-and-Ratio method and the original Shift-and-Ratio method with various value of L_{drawn} for 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)		0.08	0.09	0.11	0.20	0.30	0.50
L_{gate} (μm)		0.075	0.085	0.105	0.195	0.295	0.495
L_{eff} (μm)	Modified S&R (Vgs range of 0.1 V to 0.5 V)	0.0627	0.0712	0.1001	0.1868	0.2865	0.4813
	Modified S&R (Vgs range of 0.4 V to 0.9 V)	0.1033	0.1071	0.1219	0.1972	0.2904	0.4918
	S&R (Vgs range of 0.4 V to 0.9 V)	0.1022	0.1147	0.1290	0.1792	0.3025	0.5037

4.2.1 Carrier mobility estimation

To support our claim that the carrier mobility of the short and long channel transistor is approximately equal when the gate voltage is close to the threshold voltage, we have made an estimation of the mobility of the short and long channel transistors using the following equation:

$$\mu_{eff}(V_{gs}) = \left(\frac{L_{eff}}{W_{eff}} \right) \frac{1}{Q_{inv} R_{tot}} \quad (4.1)$$

where $R_{tot} = V_{ds}/I_{ds}$ is the total channel resistance. The inversion charge density per unit area Q_{inv} was determined from the first-order approximation $Q_{inv} = C_{ox}(V_{gs} - V_{th})$. The gate oxide capacitance per unit area, C_{ox} , was measured using a large 10 X 10 μm capacitor. The effective width, W_{eff} , can be easily determined with physical channel width of 10 μm . The L_{eff} in this case was determined by Modified Shift-and-Ratio method, the mask gate length was not use because the mobility calculated will be underestimated. The series resistance has been neglected in the mobility extraction, thus the effective mobility extracted will be slightly lower compare to that of using the intrinsic channel resistance. Another source of error in the mobility extraction comes from the approximation of Q_{inv} . The reason is that the V_{th} is not uniquely defined. In addition, Q_{inv} approaches $C_{ox}(V_{gs} - V_{th})$ at sufficiently large voltage above V_{th} , it is inaccurate near the threshold voltage and results in errors in evaluating mobility near the threshold voltage. For oxide thickness of about 10 nm, the gate voltage must exceed the threshold voltage by about 0.5 V for an error of less than 10%. To improve the accuracy of effective mobility extraction, split C - V measurement can be perform and obtain the inversion layer charge from the voltage integral of the measured split capacitance [95]. For the mobility extraction in this

discussion, the above measurement of the effective mobility is accurate enough to prove that our assumption of approximate equal mobility between short and long channel transistor at low gate voltage is true. As discussed before, V_{th} was measured using a constant-current extraction method at a constant current of $(W/L) 0.1\mu A$, where W is the channel width and L is the channel length. However, for the mobility plot in Fig. 4.7(a) and Fig. 4.7(b) the threshold voltages were reduced by 0.05V and 0.1V respectively. Such a method is good to extract the difference in the threshold voltages of the long and short channel transistors. For effective channel length measurements, it is the difference in the threshold voltages of the long and short channel transistors, which is important. However, when we used Eq. (4.1) to calculate the mobility, absolute threshold voltage has to be used. For this calculation, the threshold voltages of the long and short channel transistors have to be adjusted by the same amount such that the threshold voltage value corresponds to the transition point between sub-threshold conduction region and the low-mobility above-threshold conduction region.

Fig. 4.7: Mobility versus $V_{gs} - V_{th}$ obtained by a rough estimation method for (a) NMOS and (b) PMOS of various channel lengths.

From Fig. 4.7, it can be seen that for both the NMOS and PMOS transistors, the effective mobility in the inversion layer follows a universal curve at room temperature when plotted as a function of $V_{gs} - V_{th}$. In fact, we found that the mobility in the short channel transistor is significantly smaller than the mobility in the long channel transistor at the higher gate voltage range. At the lower gate voltage range, we found that the mobility of the short transistor is approximately equal to that of the long transistor as shown in Fig. 4.7(a) for NMOS and Fig 4.7(b) for PMOS. At low V_{gs} region, μ_c (Coulomb scattering) dominate, μ_c is the mobility controlled by channel doping impurities. In addition, for very thin oxide transistor as in this study. The electron mobility is controlled by both μ_{RCS} , the RCS mobility due to the doping impurities in the n^+ gate polysilicon and μ_{CS} , the mobility due to Coulomb scattering (CS) as mention in section 3.3. We predict that the carrier mobility is approximately equal for both the short and long channel

transistors when the gate voltage is only slightly above the threshold due to remote Coulomb scattering (RCS). The result from Fig. 4.7 is consistent with our prediction and is consistent with the work of Romanjek et al.[95]. In the work of Romanjek et al., the electron mobility measured by their new method is approximately equal for the short and long channel transistors with pocket implant when the inversion charge is small at small gate voltage. As $V_{gs} - V_{th}$ increase, the effective mobility is dominated by phonon scattering and at high $V_{gs} - V_{th}$ region the effective mobility is controlled by surface roughness. From Fig. 4.7, it can also be seen that the peak of the effective mobility gets lower as the channel length of the transistor get smaller.

4.2.2 Consistency check using SCM measurement

Recently, direct channel length measurement using cross-sectional Scanning Capacitance Microscopy (SCM) has been shown to give good results [68-70]. In this section, we compare the L_{eff} extracted by MS&R method with SCM. In addition, SCM measurement can be used as a control to generate a correction factor (CF) to further improve the accuracy of our MS&R method.

Veeco/Digital Instruments Dimension 3100 Scanning Probe Microscope was used in this experiment. The samples were cross-sectioned using a series of grinding and polishing on diamond lapping films. The final stage of polishing was done with 0.05 μ m colloidal silica. All the finished samples were obtained with a surface root mean square (rms) roughness of <0.2nm. Platinum was sputtered over the backside of the samples to ensure good ohmic contact. For thin oxide formation, we employed thermal baking at 250°C for 15 minutes under ultraviolet illumination to grow a layer of insulating oxide. Fig. 4.8 shows the SCM image of the cross-section of a PMOS transistor. From the SCM

image, channel length, L_{SCM} , of the MOSFET can be obtained by measuring the distance between the two source/drain (S/D) extensions. The SCM measurements for both NMOS and PMOS transistors are shown in Tables 4.5 and 4.6 respectively.

To extract L_{SCM} from SCM image, two effects should be considered. One is the effect of SCM dc bias voltage on the value of channel length. Another effect is the tip convolution effect. Regarding the effect of SCM dc bias voltage, we found that the channel length value was consistent within certain voltage range. These findings agree with the study of Rosenthal et al. [70]. The reason for this is that the transistors studied in this experiment have non-uniform channel doping and the effective carrier mobility is decreased in the channel region due to the halo/pocket implant. Thus, consistent channel length value can be obtained by SCM measurement.

The tip convolution effect was minimized by using XTEM (cross section Transmission Electron Microscope). During SCM scanning, topography image was acquired simultaneously with the dC/dV signal. By direct comparing the feature in topography picture with the same feature shown in TEM picture, the accuracy of the measurement can be improved. As shown in Fig. 4.9, the distance between two spacers AB were compared in TEM micrograph and topography image, the difference of the two values can be used to correct the error of SCM measurement caused by tip convolution.

Fig. 4.8: Cross-sectional scanning capacitance microscopy image of a deep-submicron PMOS transistor.

Fig. 4.9: A p-channel MOSFET: Topography image acquired during SCM scanning (left) and TEM micrograph (right). Difference between the distance AB obtained in topography image and TEM micrograph was used to correct the error of SCM measurement.

For all the $r(V_{gs})$ calculations done in the extraction of L_{eff} reported in this section, V_{gs} range used for calculation is in voltage interval step of 0.1V. This cut down the

computation time of calculating the average $r(V_{gs})$ compared to using voltage interval step of 0.01V or 0.001V without compromising the consistency of L_{eff} extracted. After the extraction of L_{eff} cross-sectional SCM was used to study short channel MOSFETs to check for the consistency of the extracted L_{eff} .

Fig. 4.10(a) shows the extracted L_{eff} in the V_{gs} range of 0.2 V to 0.5 V of both the NMOS and PMOS transistors for 0.13 μm technology. The L_{eff} extracted is fairly constant with $DL = L_{drawn} - L_{eff}$, around 0.003 μm for NMOS and around 0.018 μm for PMOS with a spread of about 1% to 4% for both NMOS and PMOS. Fig. 4.10(b) shows that the extracted L_{eff} is fairly constant with DL calculated as -0.008 μm for NMOS and 0.0165 μm for PMOS with spread of about 1.5% to 2% for NMOS and 2.5% to 10% for PMOS.

Fig. 4.10: Box plot of L_{eff} extracted using V_{gs} range from 0.2-0.5 V for both NMOS (shaded) and PMOS (unshaded) transistors for (a) 0.13 μm CMOS technology and (b) 0.1 μm CMOS technology. The magnitude of V_{ds} was set at 0.05 V.

Table 4.5 shows the effective channel length extracted by our Modified Shift-and-Ratio method (V_{gs} range of 0.2 to 0.5 V), original Shift-and-Ratio method and channel length measurement by SCM for NMOS transistors in both CMOS technologies. Both MS&R and SCM measurement generate consistent results. $L_{gate} - L_{SCM}$ is around 0.025 μm for 0.1 μm technology; L_{gate} is the physical gate length measured by XTEM. However, $L_{gate} - L_{SCM}$ is around 0.035 μm for 0.13 μm technology. $L_{gate} - L_{eff}$ is around 0.012 μm for both CMOS technologies. PMOS transistor measurements are shown in Table 4.6 $L_{gate} - L_{SCM}$ is approximately 0.043 μm and 0.046 μm for 0.1 μm and 0.13 μm technology respectively. $L_{gate} - L_{eff}$ is around 0.015 μm and 0.004 μm for 0.1 μm and 0.13 μm technology respectively. It appears to us that $L_{gate} - L_{SCM}$ is an accurate measure of gate to drain or source overlap. For NMOS transistors, the overlap is about 0.0175 μm for 0.13

Chapter 4: Results and Discussion on Modified Shift-and-Ratio Method

μm generation; it decreases to $0.0125 \mu\text{m}$ for $0.1 \mu\text{m}$ generation. For PMOS transistors, the overlap is about $0.023 \mu\text{m}$ for $0.13 \mu\text{m}$ generation; it decreases to about $0.0215 \mu\text{m}$ for $0.1 \mu\text{m}$ generation. It is obvious that the overlap is much bigger for PMOS transistors compared to NMOS transistors. This can be easily explained by the larger diffusion coefficient for boron compared to arsenic. In addition, the overlap for PMOS transistors decreases only slightly from $0.13 \mu\text{m}$ to $0.1 \mu\text{m}$ generation.

Table 4.5: A Comparison of extracted L_{eff} values for NMOS transistors by 3 different methods, namely, MS&R, S&R, and SCM methods with various values of L_{drawn} for both $0.13 \mu\text{m}$ and $0.1 \mu\text{m}$ CMOS technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)	L_{gate} (μm)	L_{SCM} (μm)	$L_{gate} - L_{SCM}$ (μm)	MS&R L_{eff} (μm)	S&R L_{eff} (μm)	$L_{gate} - \text{MS\&R}$ L_{eff} (μm)	$CF = L_{SCM}$ $/L_{eff}$
0.08	0.075	0.052	0.023	0.087	0.109	-0.012	0.597701
0.09	0.085	0.059	0.026	0.098	0.117	-0.013	0.602041
0.12	0.105	0.071	0.034	0.118	0.143	-0.013	0.601695
0.13	0.115	0.08	0.035	0.126	0.149	-0.011	0.634921

Table 4.6: A comparison of extracted L_{eff} values for PMOS transistors between MS&R, S&R and SCM measurement with different L_{drawn} for both $0.13 \mu\text{m}$ and $0.1 \mu\text{m}$ CMOS technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)	L_{gate} (μm)	L_{SCM} (μm)	$L_{gate} - L_{SCM}$ (μm)	MS&R L_{eff} (μm)	S&R L_{eff} (μm)	$L_{gate} - \text{MS\&R}$ L_{eff} (μm)	$CF = L_{SCM}$ $/L_{eff}$
0.08	0.075	0.0311	0.044	0.057	0.121	0.018	0.543221
0.09	0.085	0.0426	0.042	0.074	0.126	0.011	0.570993
0.12	0.105	0.059	0.046	0.100	0.123	0.005	0.590000
0.13	0.115	0.069	0.046	0.113	0.131	0.002	0.605464

Fig. 4.11 shows the plots of L_{eff} vs. L_{SCM} . A fairly linear relationship between the two parameters can be seen for both generations of CMOS technology. With this information, a correction factor CF can be calculated. $CF = L_{SCM} / L_{eff}$ can be used to improve the accuracy of our modified Shift-and-Ratio method. In addition, Tables 4.5 and 4.6 show that the value of L_{eff} generated by the original Shift-and-Ratio method has been greatly over-estimated.

Fig. 4.11: Plot of L_{eff} vs. L_{SCM} of the NMOS and PMOS transistors for both 0.13 μm and 0.1 μm CMOS technology.

Fig. 4.12: I-V characteristics of two NMOS transistors with the same L_{drawn} but different L_{eff} measured by our modified shift-and-ratio method with a V_{gs} range from 0.2 to 0.5 V, showing that a higher off current and a higher on current correspond to a smaller L_{eff} measured.

Fig. 4.12 shows the I-V characteristics of two NMOS transistors with the same L_{drawn} but different L_{eff} measured by modified shift-and-ratio method with a V_{gs} range from 0.2 to 0.5 V. It shows that a smaller L_{eff} measured corresponds to a higher off current, I_{off} and a higher saturation current, I_{dsat} . This is a good indication that MS&R method is correct in extracting of L_{eff} .

4.2.3 L_{eff} extraction at low bias gate voltage range

Other than using *SCM* to improve the accuracy of the L_{eff} extracted by MS&R method, it was found that the L_{eff} extracted was closer to the *SCM* measurement as the voltage ranges is decreased. In this section, the effect of different gate voltages at low bias with respect to the accuracy of extracted L_{eff} is presented. Three different V_{gs} were applied to extract the L_{eff} , these three ranges were 0.1 V to 0.5 V, 0.1 V to 0.4 V and 0.1 V to 0.3 V. The L_{eff} extracted from both NMOS and PMOS transistors with L_{drawn} of 0.12 μm , 0.13 μm and 0.15 μm for different V_{gs} ranges are shown in Fig. 4.13 and Fig. 4.14 respectively.

Fig. 4.13: L_{eff} extracted for different V_{gs} range for 0.13 μm CMOS technology NMOS transistors. V_{ds} was biased at 0.05 V.

Fig. 4.14: L_{eff} extracted for different V_{gs} range for 0.13 μm CMOS technology PMOS transistors. V_{ds} was biased at 0.05 V.

Fig. 4.13 shows that the extracted L_{eff} for NMOS transistor get smaller as V_{gs} ranges decreased. Similarly, PMOS transistors exhibit the same trend as shown in Fig. 4.14. Similar trend was observed for both NMOS and PMOS transistors for 0.1 μm technology. The effective channel length extracted in the V_{gs} range of 0.1 V to 0.3 V was fairly constant with $DL = L_{drawn} - L_{eff}$, around 0.032 μm for NMOS and around 0.051 μm for PMOS for the 0.13 μm technology. For 0.1 μm technology, DL calculated is around 0.018 μm for NMOS and around 0.033 μm for PMOS. The extracted L_{eff} in the V_{gs} range of 0.1 V to 0.3 V was closer to that measured by *SCM*.

In the next section, further comparison between the L_{eff} extracted by MS&R and capacitance method is made and the implication of extracting L_{eff} at lower V_{gs} ranges will be discussed.

4.2.4 L_{eff} correlation with I_{off}

In this section, L_{eff} extracted using capacitance method [46] was performed and compared with that of our new MS&R method using MOS transistors fabricated by 0.11 μm CMOS technology on 8-inch silicon wafers with the physical gate oxide thickness smaller than 2 nm. Further to that, the correlation between the extracted L_{eff} and I_{off} will be presented.

The relationship between L_{eff} and the logarithm of I_{off} was used to check the validity of the L_{eff} extraction method. It was observed that the on-current (I_{on}) and the logarithm of the off-current ($\log(I_{off})$) of modern submicron MOS transistors tend to follow a very good linear relationship as shown in Fig. 4.15(a) and 4.15 (b) [109]. As shown in Fig. 4.15(a), L_{eff} can be seen as a hidden parameter in the I_{on} - $\log I_{off}$ plot. As shown in Fig. 4.15(b), L_{eff} can be seen as a hidden parameter in the I_{on} - $\log I_{off}$ plot. Instead of plotting I_{on} against $\log I_{off}$, both I_{on} and $\log I_{off}$ can be plotted against drain induced barrier lowering DIBL.

Fig. 4.15: (a) Variation of L_{eff} in I_{on} - $\log I_{off}$ relationship (b) Variation of DIBL in I_{on} - $\log I_{off}$ relationship.

Experimentally, our data show that I_{on} has a very good linear relationship with (DIBL) as shown in Fig. 4.16. Similarly, $\log I_{off}$ has a very good linear relationship with DIBL as shown in Fig. 4.17. Since both I_{on} and $\log I_{off}$ vary in a linear manner with DIBL, I_{on} can be deduced to be linearly related to $\log I_{off}$ as reported in [109]

Fig. 4.16: I_{on} versus DIBL for both NMOS and PMOS transistors

Fig. 4.17: Logarithm of I_{off} versus DIBL for both NMOS and PMOS transistors

By measuring the I_{off} or I_{on} of a MOSFET transistor and correlate it to the extracted L_{eff} , it can be use as an simple and easy way to validate the extraction method, Here, the use of $\log I_{off}$ rather than I_{on} was chosen for comparison. The Correlation Coefficient (CC) is a measurement of the strength of the linear association between two variables. The correlation coefficient always takes a value between -1 and 1, with 1 or -1 indicating perfect correlation. (Perfect correlation means that all points would lie along a straight line in this case). A positive correlation indicates a positive association between the variables (increasing values in one variable correspond to increasing values in the other variable), while a negative correlation indicates a negative association between the variables (increasing values is one variable correspond to decreasing values in the other variable). A correlation value close to 0 indicates no association between the variables.

Since the formula for calculating the correlation coefficient standardizes the variables, changes in scale or units of measurement will not affect its value. For this reason, the correlation coefficient is often more useful than a graphical depiction in determining the strength of the association between two variables.

Chapter 4: Results and Discussion on Modified Shift-and-Ratio Method

The square of the correlation coefficient, R^2 , is a useful value in linear regression. This value represents the fraction of the variation in one variable that may be explained by the other variable. Thus, if a correlation of 0.8 is observed between two variables (say, height and weight, for example), then a linear regression model attempting to explain either variable in terms of the other variable will account for 64% of the variability in the data.

Table 4.7: A comparison of extracted L_{eff} values for NMOS transistors measured by MS&R, capacitance and S&R methods with various values of L_{drawn} for 0.11 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)	L_{gate} (μm)	MS&R L_{eff} (μm) (0.1 to 0.5V)	MS&R L_{eff} (μm) (0.1 to 0.4V)	MS&R L_{eff} (μm) (0.1 to 0.3V)	CV L_{eff} (μm)	S&R L_{eff} (μm) (0.4 to 0.8V)
0.12	0.065	0.0751	0.0679	0.0568	0.0519	0.0793
0.13	0.075	0.0768	0.0759	0.0742	0.0658	0.0852
0.14	0.085	0.0885	0.0833	0.0757	0.0789	0.0999

Table 4.8: A comparison of extracted L_{eff} values for PMOS transistors between MS&R, capacitance and S&R method with different L_{drawn} for 0.11 μm CMOS technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)	L_{gate} (μm)	MS&R L_{eff} (μm) (0.1 to 0.5V)	MS&R L_{eff} (μm) (0.1 to 0.4V)	MS&R L_{eff} (μm) (0.1 to 0.3V)	CV L_{eff} (μm)	S&R L_{eff} (μm) (0.4 to 0.8V)
0.12	0.07	0.0743	0.0685	0.0577	0.0584	0.0829
0.13	0.08	0.0811	0.0779	0.0702	0.0699	0.0979
0.14	0.09	0.0910	0.0898	0.0844	0.0818	0.1189

Table 4.7 and Table 4.8 show that the L_{eff} extracted by MS&R method is close to that extracted by capacitance method whereas S&R method over estimated the L_{eff} for both

NMOS and PMOS transistors. Consistence to other CMOS technologies under study, the extracted L_{eff} decreases as the scan V_{gs} range decreases.

The leakage currents (I_{off}) of the CMOS devices were measure to check for the relation to the extracted L_{eff} . Fig. 4.18 shows the correlation between L_{eff} and I_{off} for NMOS device with L_{drawn} of 0.12 μ m. It shows a linear relationship where I_{off} increase as L_{eff} gets smaller, for both the MS&R and capacitance method.

Fig. 4.18: Plot of I_{off} vs. L_{eff} for NMOS transistors fabricated by 0.11 μ m CMOS technology. L_{eff} was extracted by MS&R method (0.1-0.5 V V_{gs} range) and also the capacitance method. (Note: CC is correlation coefficient.)

Figure 4.18 shows that the correlation coefficients between L_{eff} and I_{off} generated by both MS&R and capacitance L_{eff} extraction method. From figure 4.18, it was found that MS&R method has better correlation than capacitance method. One explanation for

L_{eff} extracted by capacitance method having worse correlation coefficients is that the test structure use for L_{eff} extraction by capacitance method is a special structure which is different from a conventional transistor device.

Fig. 4.19: Plot of I_{off} vs. L_{eff} for NMOS transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method at 3 different V_{gs} ranges and also by the S&R method for one V_{gs} range. (Note: CC is correlation coefficient.)

Comparing the correlation coefficients generated by different L_{eff} extraction method, it was found that MS&R method has better correlation than capacitance method and S&R method. It can also be seen from Fig. 4.19 that as the scan V_{gs} ranges decrease for the MS&R method, the correlation gets better. This is an indication that, as the V_{gs} range decreases, the mobility between the short and the long channel gets more equal resulting in a better correlation.

The leakage current (I_{off}) of the PMOS devices was measured to check for the relation to the extracted L_{eff} . Fig. 4.20 shows the correlation between L_{eff} and I_{off} for PMOS device with L_{drawn} of 0.12 μ m. It shows a linear relationship where I_{off} increase as L_{eff} gets smaller, for both the MS&R and capacitance method.

Fig. 4.20: Plot of I_{off} vs. L_{eff} for PMOS transistors fabricated by 0.11 μ m CMOS technology. L_{eff} was extracted by MS&R method (0.1-0.5 V V_{gs} range) and the capacitance method. (Note: CC is correlation coefficient.)

Similarly for PMOS transistor, it was found that the magnitude of correlation coefficient generated by MS&R method is closer to 1 than the capacitance method and the S&R method. From Fig. 4.21, it can be seen that as the scan V_{gs} ranges decrease for MS&R method, the magnitude of the correlation coefficient gets closer to 1. Hence it is a good indication that the L_{eff} extracted is consistent and the assumption that m_{eff} between

short channel and long channel transistor is approximately equal in the subthreshold region is valid.

Fig. 4.21: Plot of I_{off} vs. L_{eff} for PMOS L_{eff} (mm) transistors fabricated by 0.11 μm CMOS technology. L_{eff} was extracted by MS&R method for 3 different V_{gs} ranges and also by the S&R method for one V_{gs} range. (Note: CC is correlation coefficient.)

Judging from the correlation coefficient generated by the capacitance method, it shows that this method had a disadvantage as the test structure use for L_{eff} extraction is different from the actual MOS transistor.

Fig. 4.22: Plot of I_{off} vs. L_{eff} for NMOS and PMOS transistors fabricated by 90 nm CMOS technology with L_{gate} of 60nm. L_{eff} was extracted by MS&R method (0.1-0.3 V V_{gs} range), (Note: CC is correlation coefficient.)

The leakage currents (I_{off}) of the CMOS devices for 90nm technology were measured to check for the relation to the extracted L_{eff} . Fig. 4.22 shows the correlation between L_{eff} and I_{off} for both NMOS and PMOS device with L_{gate} of 60 nm. It shows a linear relationship where I_{off} increase as L_{eff} gets smaller.

4.3 Results and discussion II

To eliminate the big difference in the physical properties between the long and short channel transistors, a short channel transistor with nearly the same channel length is used as the reference transistor instead of a long channel transistor when extracting the L_{eff} . The relative changes in L_{eff} , $\%L_{eff}$, compared to a reference transistor will be presented in this section.

Table 4.9: Inline poly Develop Image Critical Dimension (DICD)/ Final Image Critical Dimension FICD results (DICD and FICD were measured in μm .)

	BARC Over-etch trimming							
	Wafer 1 (0%)		Wafer 2 (25%)		Wafer 3 (50%)		Wafer 4 (75%)	
Site	DICD	FICD	DICD	FICD	DICD	FICD	DICD	FICD
1	0.1394	0.1023	0.1392	0.0960	0.1433	0.0943	0.1377	0.0901
2	0.1387	0.1046	0.1380	0.1007	0.1440	0.0958	0.1400	0.0923
3	0.1392	0.1041	0.1377	0.0966	0.1359	0.0912	0.1321	0.0911
4	0.1349	0.1071	0.1356	0.0946	0.1329	0.0937	0.1353	0.0897
5	0.1358	0.1040	0.1361	0.0985	0.1379	0.0942	0.1309	0.0888
6	0.1344	0.1005	0.1366	0.0997	0.1400	0.0919	0.1345	0.0897
7	0.1387	0.1045	0.1359	0.1007	0.1371	0.0927	0.1380	0.0909
8	0.1376	0.1023	0.1395	0.0979	0.1342	0.0939	0.1398	0.0918
9	0.1379	0.1031	0.1395	0.1003	0.1311	0.0900	0.1384	0.0913
AVE	0.1374	0.1036	0.1376	0.0983	0.1374	0.0931	0.1363	0.0906

Table 4.9 shows the inline DICD and FICD measurement of four different wafers with different BARC (Bottom Anti-Reflective Coating) over-etch trimming. BARC is a bottom anti-reflective layer coating for use on highly reflective surfaces in the semiconductor industry. It reduces light reflectance during lithography, a problem that can distort line widths in integrated circuits pattern printing. The BARC, when applied to a substrate, lowers the reflectance of light back into the photoresist to less than 1 percent. It does so by absorbing the radiation that has passed through the photoresist and dissipating the energy as heat. Doing so will help shrink line widths to as narrow as 0.08

microns (μm). After the photoresist was developed, the photoresist went through an anisotropic etch by O_2/CF_4 to trim the sidewall of the photoresist. By means of controlling the etch time, the width of the photoresist can be trimmed to a smaller dimension before poly-Si etch. DICD (Develop Image Critical Dimension) is a measurement of the dimension of the photoresist for poly-Si etch after the photoresist was developed. FICD (Final Image Critical Dimension) is a measurement of the dimension of the poly-si after it was etched. Both DICD and FICD were measured inline by a CD-SEM (Critical Dimension Scanning Electron Microscope) thus the measurement was done while the processing wafer is still under fabrication. Please refer to Appendix A for the details of the poly-si etch process.

Fig 4.23 shows the box plot of the measured FICD against various percentages of BARC over-etch trimming. Wafer 1 with 0% trimming has a physical gate length of about $0.1036 \mu\text{m}$. Wafer 4 with the highest over-etch trimming of 75% has a physical gate length of about $0.0906 \mu\text{m}$, that is about $0.01 \mu\text{m}$ shorter than the control wafer.

Fig 4.23: Box plot of the measured FICD against various percentages of BARC over-etch trimming.

Our Modified Shift-and-Ratio method was applied to the short channel transistor with channel length of 0.13 μm and channel width of 20 μm for all the four wafers. A total of nine sites on each wafer was measured with site number 6 (center) of wafer 1 chosen to be the reference transistor for all the calculation of the relatively change in L_{eff} , ΔL_{eff} . The following figures show the plot of I_{dsat} vs. the extracted ΔL_{eff} with respect to the reference transistor of wafer 1 for all four wafers. I_{dsat} vs. FICD plot for each wafer is shown and the correlation is compared to that between the I_{dsat} vs. the extracted relative change in L_{eff} .

Fig.4.24: I_{dsat} vs. relative change in L_{eff} plot of wafer 1 (0 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

Fig. 4.25: I_{dsat} vs. FICD plot of wafer 1 (0 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

From Fig. 4.24 and Fig. 4.25, we can see that our calculation of the L_{eff} generates a better correlation with I_{dsat} compared to FICD measurement.

Fig. 4.26: I_{dsat} vs. relative change in L_{eff} plot of wafer 2 (25 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

Fig. 4.27: I_{dsat} vs. FICD plot of wafer 2 (25 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

Judging from Fig.4.26 and Fig. 4.27, we can see that for wafer 2, our calculation of the L_{eff} does not generate a better correlation with I_{dsat} compared to FICD measurement.

Fig. 4.28: I_{dsat} vs. relative change in L_{eff} plot of wafer 3 (50 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

Fig.4.29: I_{dsat} vs. FICD plot of wafer 3 (50 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

From Fig.4.28 and Fig. 4.29, we can see that for wafer 3, our calculation of the ΔL_{eff} does not generate a better correlation with I_{dsat} compared to FICD measurement.

Fig. 4.30: I_{dsat} vs. relative change in L_{eff} plot of wafer 4 (75 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

Fig. 4.31: I_{dsat} vs. FICD plot of wafer 4 (75 % over-etch) (Note: R^2 is the square of the correlation coefficient.)

From Fig. 4.30 and Fig. 4.31, we can see that for wafer 4, our calculation of the ΔL_{eff} generates a better correlation with I_{dsat} compared to FICD measurement.

Comparing the result across four wafers, I_{dsat} vs. ΔL_{eff} plot generates a R^2 value ranging from 0.004 to 0.45. R^2 values generated from I_{dsat} Vs FICD plot range from 0.0001 to 0.14. Both sets of data show that the correlation between I_{dsat} and L_{eff} or FICD was not very consistent from wafer to wafer, with FICD having the worse correlation. This is due to a certain degree of measurement error and the spread can be seen from Fig. 4.23. Therefore, it was not an efficient tool to measure very small changes in poly length within wafer. The following two figures show the correlation of FICD with I_{dsat} and ΔL_{eff} with I_{dsat} plotted for all four wafers.

Fig. 4.32: I_{dsat} vs. relative change in L_{eff} plot for all the four wafers (Note: R^2 is the square of the correlation coefficient.)

Fig. 4.33: I_{dsat} vs. FICD plot for all the four wafers (Note: R^2 is the square of the correlation coefficient.)

Our modified Shift-and-Ratio method was applied to the four wafers to extract the L_{eff} . Similarly, nine sites were measured for each wafer using a transistor with channel length and width of 20μm as reference transistor. The extracted L_{eff} is plotted against I_{dsat} as shown in Fig. 4.34.

As shown in Fig. 4.32, Fig.4.33 and Fig. 4.34, L_{eff} extracted using our Modified Shift-and-Ratio method generates the best correlation compared to the FICD measurement and relative change in the L_{eff} calculation.

Fig. 4.35: MS&R L_{eff} vs. FICD measurement of NMOS devices with various physical gate lengths.

Fig. 4.35 shows the correlation between L_{eff} extracted by MS&R and FICD measurement, L_{eff} measured by our MS&R method is able to track the in-line FICD measurement quite closely.

4.4 Chapter summary

In summary, our Modified Shift-and-Ratio methods for L_{eff} extraction in deep-submicron MOS transistors were developed and verified on advanced 0.13 μm , 0.11 μm , 0.1 μm and 90 nm MOS transistors with halo/pocket implant.

Experimentally, we found that our Modified-Shift-and-Ratio method tends to yield more reasonable values than the original shift-and-ratio method. This is a strong indication that our direct measurement of threshold voltages may be better than shifting and matching the $S(V_{gs})$ vs. V_{gs} curves for the device under test and the reference device in the original shift-and-ratio method. In addition, our proposed method is also much simpler and thus much more convenient for routine electrical testing of deep-submicron MOS transistors in the industry. Our method is also less sensitive to measurement noise, especially when the gate voltage is close to the threshold voltage. In this context, we have successfully used a modified shift-and-ratio method to extract the effective channel length L_{eff} for MOS transistors fabricated by state-of-the-art CMOS technology. In addition, we have studied very short channel MOSFETs using cross-sectional SCM. Comparing the L_{eff} extracted using our modified shift-and-ratio method, original S&R method and L_{SCM} measured using SCM, all three measurements show the consistency with respect to L_{gate} . However, the MS&R method yields a L_{eff} nearer to the SCM measurement than the original shift-and-ratio method. Furthermore, L_{eff} extracted using MS&R method is comparable to L_{eff} measured using the capacitance method. In addition, good correlation between L_{eff} and I_{off} is shown with a higher off current and saturation current corresponding to a smaller L_{eff} . Furthermore, our MS&R method is non-destructive whereas cross-sectional SCM is a destructive technique. However, SCM is able to provide direct physical imaging of MOSFET and thus can be used as a verification tool

for our MS&R method. To improve the accuracy of MS&R, a correction factor (CF) can be generated by using the channel length value extracted by SCM.

A second technique that is much simpler was proposed to extract the relatively change in L_{eff} of MOSFETs. This may be more convenient for routine electrical testing of deep-submicron MOS transistors in the industry. This technique is simple as it is easy to implement, it needs only the I_d - V_g measurement for the calculation of the relative change in L_{eff} . It is fast to test, once the reference transistor Id-Vg characteristic is defined, only one measurement needs to be done for L_{eff} ratio calculation on each site. Effectively, this approach reduces testing time to half of that for the conventional method.

Although it generates a better modeling of the I_{ds} - L_{eff} relationship compared to FICD measurement devices with 0% and 75% poly over etch but it is not very consistent when it was applied to more than four wafers. The above experimental results indicate that our MS&R method may not be sensitive enough to detect a small change in channel length due to process variation. In fact, through our experiment, drain-induced-barrier-lowering (DIBL) was found to be more sensitive to process variation, therefore DIBL can be used to monitor channel length variation due to process variation. However, additional work has to be done to correlate DIBL with L_{eff} .

CHAPTER FIVE

Results and Discussion on Subthreshold Regime-Based Method

This chapter presents the results generated by our novel method for accurate L_{eff} extraction based on the sub-threshold regime. This new method was tested on both n-channel Metal-Oxide-Silicon (NMOS) and p-channel Metal-Oxide-Silicon (PMOS) transistors fabricated by state-of-the-art Complementary Metal-Oxide-silicon (CMOS) technology. The extracted L_{eff} by our subthreshold Regime-Based method was compared with the original S&R method and also the capacitance based methods. In addition, we have studied the cross-sections of the MOSFETs using SCM and by comparing our extracted L_{eff} measurement with that of the SCM measurement, we are able to show the consistency of our L_{eff} extraction method. These results provide strong support to our theory, which predicts that the carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is below the threshold voltage.

5.1 Result and discussion III

The Subthreshold Regime-Based L_{eff} extraction method was applied to both NMOS and PMOS transistors fabricated using 0.13 μm , 0.11 μm , 0.1 μm and 90nm CMOS technology and the results will be presented in this chapter. Before the extraction of L_{eff} , it is necessary to determine the threshold voltage; V_{th} .

It is well known that V_{th} is an important MOSFET parameter which directly determines the turn-on and turn-off of the MOS transistors. Therefore, its accurate estimation is a key concern for device characterization and modeling. However, V_{th} is a voltage that is not uniquely defined. There are several different techniques [102-108,110-112]; however, the extracted values for the V_{th} vary from method to method. Each method has its advantages and disadvantages, a detail discussion can be found in Appendix B of the thesis. In this part of the study, various methods of extracting V_{th} were performed and the resulting L_{eff} extracted was evaluated.

A total of nine different V_{th} extraction methods were performed, including the constant current (CC) method [102], maximum transconductance (MAX GM) method [102], transconductance change (TC) method [103], match point (MP) method [104], second derivative of the logarithm of drain current (SDL) method [105], transconductance linear extrapolation (GMLE) method [106], maximum G-function (MAX G-FUNCTION) method [107], normalized mutual integral difference (NMID) method [108] and the “Third Derivative of Drain-Source Current” (TD) method [109].

Table 5.1 and Table 5.2 show the V_{th} values extracted by various methods with various channel lengths for 2 n-MOS transistors under test fabricated using 0.11 μm CMOS technology. All the extracted value of V_{th} demonstrates the reverse short channel effect when plotted against the channel lengths [113-115].

Chapter 5: Results and Discussion on Subthreshold regime-Base Method

Table 5.1: A comparison of threshold voltage values extracted by various methods for n-MOS transistor A with various channel lengths.

Transistor A	CC	Match point	GMLE	TC	Max GM	SDL	TD	NMID	Max G function
L_{drawn} (μm)	Extracted V_{th} (V)								
0.12	0.23	0.24	0.27	0.34	0.32	0.32	0.27	0.34	0.32
0.13	0.28	0.24	0.29	0.33	0.35	0.33	0.29	0.34	0.35
0.14	0.29	0.25	0.37	0.36	0.36	0.38	0.35	0.35	0.37
10	0.26	0.22	0.25	0.30	0.32	0.30	0.27	0.30	0.32

Table 5.2: A comparison of threshold voltage extracted by various methods for n-MOS transistor B with various channel lengths.

Transistor B	CC	Match point	GMLE	TC	Max GM	SDL	TD	NMID	Max G function
L_{drawn} (μm)	Extracted V_{th} (V)								
0.12	0.25	0.23	0.28	0.33	0.33	0.33	0.27	0.33	0.33
0.13	0.28	0.23	0.30	0.41	0.35	0.38	0.29	0.34	0.35
0.14	0.29	0.25	0.29	0.32	0.36	0.35	0.31	0.34	0.36
10	0.25	0.22	0.25	0.30	0.32	0.32	0.26	0.29	0.32

V_{th} extracted by MAX G-FUNCTION method shows the highest value while the V_{th} extracted by MP method is lowest. The Subthreshold Regime-Based L_{eff} extraction method was then applied to the n-MOSFET transistor using the extracted V_{th} with various methods. Fig. 5.1 shows the extracted delta L ($\Delta L = L_{drawn} - L_{eff}$) value plotted against different L_{drawn} generated by different V_{th} values. It was found that V_{th} extracted by NMID method generates the most consistent ΔL of about 0.05 μm . This technique is insensitive to the parasitic source and drain resistance while sensitive to the channel length variation.

V_{th} extracted by GMLE method generate a reasonable ΔL between 0.02 μm to 0.06 μm . The ΔL values generated by the others V_{th} extraction methods were either not consistent or unrealistic. There are several reasons for this inconsistency. For examples, the LE method is the most extensively employed by researchers, but it presents the disadvantage of being dependent of the series resistances of the MOSFET; The SDL and TC method developed to avoid the dependence on the series resistances, is highly sensitive to noise in the measurements because the second derivative is equivalent to a high pass filter.

Fig. 5.1: Plot of ΔL generated by the Subthreshold Regime-Based method using V_{th} value extracted by different threshold voltage extraction methods with different channel length of n-MOS transistors fabricated using 0.11 μm CMOS technology.

With V_{th} extracted using the NMID method, the Subthreshold Regime-Based L_{eff} extraction method was applied to both NMOS and PMOS transistors fabricated using 0.13

μm , $0.11 \mu\text{m}$ and $0.1 \mu\text{m}$ CMOS technology. One difficulty to note in extracting L_{eff} for a PMOS transistor is the determination of the *sub-threshold swing*, S . Due to the effect of super steep retrograde well (SSRW) use to improve PMOS transistor performance [116], the parameters may not be a constant over the range of the I-V sub-threshold curve. To overcome this difficulty, a smaller range of the I-V sub-threshold curve is used to calculate the parameter S .

The extracted L_{eff} are shown in Table 5.3 for three different short channel length NMOS transistors of $0.11 \mu\text{m}$ CMOS technology. The L_{eff} extracted using Subthreshold Regime-Based method was fairly constant with $L = L_{drawn} - L_{eff} = 0.074 \mu\text{m}$. Similarly as shown in Table 5.4, L_{eff} extracted using Subthreshold Regime-Based method was fairly constant with $L = L_{drawn} - L_{eff} = 0.058 \mu\text{m}$ for p-channel MOS transistors. Hence it is a good indication that the L_{eff} extracted is consistent and our assumption that the m_{eff} between short channel and long channel transistor is approximately equal at subthreshold region is valid. In addition, L_{eff} extracted using the capacitance method [46] was performed and compared with that using the new method, L_{eff} extracted using the new method was slightly higher than that by the capacitance method consistently for PMOS transistors and comparable for n-channel MOS transistors.

Table 5.3: A comparison of extracted L_{eff} values for NMOS transistors between Subthreshold Regime-Based, MS&R, capacitance and S&R method with different L_{drawn} for $0.11 \mu\text{m}$ technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (μm)	L_{gate} (μm)	Subthreshold L_{eff} (μm)	MS&R L_{eff} (μm) (0.1 to 0.3V)	CV L_{eff} (μm)	S&R L_{eff} (μm) (0.4 to 0.8V)
0.12	0.065	0.0458	0.0568	0.0519	0.0793
0.13	0.075	0.0607	0.0742	0.0658	0.0852
0.14	0.085	0.0663	0.0757	0.0789	0.0999

Chapter 5: Results and Discussion on Subthreshold regime-Base Method

Table 5.4: A comparison of extracted L_{eff} values for PMOS transistors between Subthreshold Regime-Based, MS&R, capacitance and S&R method with different L_{drawn} for 0.11 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (mm)	L_{gate} (mm)	Subthreshold L_{eff} (mm)	MS&R L_{eff} (mm) (0.1 to 0.3V)	CV L_{eff} (mm)	S&R L_{eff} (mm) (0.4 to 0.8V)
0.12	0.07	0.0616	0.0577	0.0584	0.0829
0.13	0.08	0.0720	0.0702	0.0699	0.0979
0.14	0.09	0.0826	0.0844	0.0818	0.1189

Table 5.5: A comparison of extracted L_{eff} values for NMOS transistors among Subthreshold Regime-Based, MS&R, S&R, and SCM measurement with different L_{drawn} for both 0.13 μm and 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM

L_{drawn} (mm)	L_{gate} (mm)	L_{SCM} (mm)	Subthreshold L_{eff} (mm)	MS&R L_{eff} (mm)	S&R L_{eff} (mm)
0.08	0.075	0.052	0.055	0.087	0.109
0.09	0.085	0.059	0.069	0.098	0.117
0.12	0.105	0.071	0.082	0.118	0.143
0.13	0.115	0.08	0.093	0.126	0.149

Table 5.6: A comparison of extracted L_{eff} values for PMOS transistors between Subthreshold Regime-Based, MS&R, S&R and SCM measurement with different L_{drawn} for both 0.13 μm and 0.1 μm technology. L_{gate} is the physical gate length measured by XTEM.

L_{drawn} (mm)	L_{gate} (mm)	L_{SCM} (mm)	Subthreshold L_{eff} (mm)	MS&R L_{eff} (mm)	S&R L_{eff} (mm)
0.08	0.075	0.0311	0.056	0.057	0.121
0.09	0.085	0.0426	0.064	0.074	0.126
0.12	0.105	0.059	0.072	0.100	0.123
0.13	0.115	0.069	0.091	0.113	0.131

The L_{eff} extracted using Subthreshold Regime-Based method was fairly constant with
 $L = L_{drawn} - L_{eff} = 0.0185\mu\text{m}$ for both NMOS and PMOS transistors for 0.1 μm technology
. Similarly as shown in Table 5.5 and Table 5.6 for 0.13 μm technology, L_{eff} extracted using

the Subthreshold Regime-Based method was fairly constant with $L = L_{drawn} - L_{eff} = 0.022$ $L = L_{drawn} - L_{eff} = 0.029$ μm for NMOS transistors and PMOS transistors respectively. Comparing the extracted L_{eff} of the $0.11\mu\text{m}$ CMOS technology with the capacitance method and the L_{eff} extracted for $0.1\mu\text{m}$ and $0.13\mu\text{m}$ CMOS technologies with that measured by *SCM*, it was found that for PMOS transistors, the extracted L_{eff} was slightly higher than *SCM* measurement whereas NMOS transistors show comparable result with *SCM* measurement.

The leakage current (I_{off}) of the CMOS devices were measured to check for the relation to extracted L_{eff} . Fig. 5.2 shows the correlation between L_{eff} and I_{off} for both NMOS and PMOS device of $0.11\mu\text{m}$ CMOS technology. It shows a linear relationship where I_{off} increase as L_{eff} gets smaller. It was found that the correlation coefficient is about -0.568 and -0.763 for NMOS and PMOS transistors, respectively.

Fig. 5.2: Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.11 mm CMOS technology with L_{drawn} of 0.12 mm

Similarly, the leakage current (I_{off}) of the devices for 0.13 μm and 0.1 μm technology were measured to check for the relation to the extracted L_{eff} . Fig. 5.3 shows the correlation between L_{eff} and I_{off} for both NMOS and PMOS transistors for 0.13 μm technology and Fig. 5.4 for 0.1 μm CMOS technology. It shows a linear relationship where I_{off} increase as L_{eff} gets smaller for both technologies.

Fig. 5.3: Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.13 μm CMOS technology with L_{drawn} of 0.12 μm , 0.13 μm and 0.15 μm .

Fig. 5.3 shows a linear relationship where I_{off} increase as L_{eff} gets smaller. It was found that the correlation coefficient is about -0.589 and -0.628 for NMOS and PMOS transistors respectively.

Fig. 5.4: Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 0.1 μm CMOS technology with L_{drawn} of 0.08 μm, 0.09 μm and 0.11 μm.

Fig. 5.4 shows a linear relationship where I_{off} increase as L_{eff} gets smaller. It was found that the correlation coefficient is about -0.852 for both NMOS and PMOS transistors.

Fig. 5.5: Plot of I_{off} vs. Subthreshold Regime-Based L_{eff} for both NMOS and PMOS transistors fabricated by 90 nm CMOS technology with L_{gate} of 60 nm.

Fig. 5.5 shows a linear relationship where I_{off} increase as L_{eff} gets smaller. It was found that the correlation coefficient is about -0.341 for NMOS and -0.620 for PMOS transistors.

5.2 Chapter summary

In summary, our Subthreshold Regime-Based method for L_{eff} extraction in deep-submicron MOS transistors was proposed and verified on advanced 0.13 μm , 0.11 μm and 0.1 μm MOS transistors with halo/pocket implant.

It overcomes the limit of conventional L_{eff} extraction algorithms that are based on the assumption that the mobility is independent of gate length. We have proposed that the

carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is in the sub-threshold region. We have proved our theory from the successful application of our new method in sub-micron CMOS technology to yield consistent and reasonable L_{eff} values.

It is also much easier and simpler to implement than the capacitance method, as it does not require special test structures to get sufficient capacitance signal for L_{eff} measurement. In addition, this new method is done at small gate voltage while the capacitance method involves measurement at higher gate voltage where gate oxide leakage can cause measurement errors.

Comparing the L_{eff} extracted using the subthreshold slope method and L_{eff} measured using capacitance method, both measurements show the consistency with respect to L_{gate} . Furthermore, the L_{eff} extracted using the subthreshold slope method is very close to that measured by *SCM*. In addition, good correlation between L_{eff} and I_{off} is shown with a higher off current corresponding to a smaller L_{eff} .

CHAPTER SIX

Conclusions and Recommendations

6.1 Conclusions

This chapter briefly summarizes the work presented in this thesis. The focus of this thesis is on the measurement of the effective channel length L_{eff} of state-of-the-art n-channel and p-channel MOS transistors with halo/pocket implant. Finally, some suggestions for future work will be presented.

6.1.1 A review of goal and achievement

In Chapter One, the necessity of using the halo/pocket implant in state-of-the-art MOS transistors was explained. In addition, it has been pointed out that the presence of the halo/pocket implant tends to make carrier mobility in the channel region smaller for the short transistor than that for the long transistor. Thus the carrier mobility in state-of-the-art MOS transistors is a function of the gate length. However, for conventional channel resistance based family of L_{eff} measurement techniques, the basic assumption is that the carrier mobility is a constant independent of the gate length. Thus it is a very challenging task to measure L_{eff} accurately for state-of-the-art MOS transistors with halo/pocket implant. The objective of this thesis work, as explained in Chapter 1, is to

Chapter 6: Conclusions and Recommendations

modify existing L_{eff} extraction methods or to develop a totally new method suitable for state-of-the-art MOS transistors with halo/pocket implant used in CMOS technology.

In Chapter Two, various existing L_{eff} extraction methods were reviewed. For non-destructive methods, there are two families of methods. Channel resistance based methods will run into difficulties as explained in Chapter 1. Capacitance based methods require special test structures. Cross-sectional SCM is a destructive technique.

In Chapter Three, the theory of a method, which has been developed during this PhD project and is totally different from the various methods explained in Chapter 2, was explained in detail. This method is based on the sub-threshold I-V characteristics.

In Chapter Four, the Modified Shift-and-Ratio method is applied on advanced 0.13 μm , 0.11 μm and 0.1 μm MOS transistors. This method, which is modified based on the proposed theory as presented in Chapter two, generate accurate and consistent results compared to that measured by Scanning Capacitance Microscopy (SCM) and Capacitance L_{eff} extraction method.

In Chapter Five, this novel method of L_{eff} extraction based on the subthreshold regime was verified on advanced 0.13 μm , 0.11 μm and 0.1 μm CMOS technology. Experimentally data from SCM measurement and L_{eff} measured by the capacitance based method prove that applying this novel method to state-of-the-art MOS transistors fabricated by deep-submicron CMOS technology yields L_{eff} values which are consistent and reasonable.

Furthermore, a new insight of the physics of MOS transistors operating at low gate voltage in the sub-threshold regime was discovered during this PhD project: the carrier mobility in the sub-threshold regime is dominated by remote Coulomb scattering (RCS) mechanism. Experimental results presented in this thesis provide a strong support

to our theory, which further predicts that the carrier mobility is approximately equal for both the short and long channel transistors when the gate voltage is below the threshold voltage.

In summary, this PhD project overcomes the very challenging task to measure L_{eff} accurately for state-of-the-art MOS transistors with halo/pocket implant. In addition, new insight of MOS device physics has also been achieved.

6.1.2 List of publications

A list of papers resulting from this PhD project are shown below:

Journal papers:

1. C.-W. Eng, W.-S. Lau, D. Vigar and J. Y.-M. Lee, "An Improved Shift-and-Ratio Effective Channel Length Extraction Method for Metal Oxide Silicon Transistors with Halo/Pocket Implants," *Jpn. J. Appl. Phys.* Vol. 42, Part 1, No. 5A, p.2621, May 2003.
2. K.-S. See, W.-S. Lau, H. Liao, C.-W. Eng, K.-M. Tee, E. K.-B. Quek, K.-C. Tee and L.-H. Chan, "Effective Channel Length Shortening and Mobility Increase of p Channel Metal Oxide Semiconductor Transistors Resulting in Higher Drive Current Using Short Source–Drain Diffusion Length," *Jpn. J. Appl. Phys.*, Vol. 43, No. 3, p.925, 2004.
3. C.-W. Eng, W.-S. Lau, Y.-Y. Jiang, D. Vigar, K.-C. Tee, L. Chan, V. S.-W. Lim and A. Trigg, "Improving the Accuracy of Modified Shift-and-Ratio Channel Length Extraction Method Using Scanning Capacitance Microscopy," *Jpn. J. Appl. Phys.*, Vol. 43, No. 4B, p.1869, 2004.

Chapter 6: Conclusions and Recommendations

4. C. W. Eng, W. S. Lau, D. Vigar, S. S. Tan, and L. Chan, ‘Effective channel length measurement of metal-oxide-semiconductor transistors with pocket implant using the subthreshold current-voltage characteristics based on remote Coulomb scattering,” *Appl. Phys. Lett.*, Vol. 87, article number 153510, 2005.
5. W.S. Lau, Peizhen Yang, C.W. Eng, V. Ho, C.H. Loh, S.Y. Siah, D. Vigar and L. Chan, “A study of the linearity between I_{on} and $\log I_{off}$ of modern MOS transistors and its application to stress engineering,” accepted for publication in *Microelectronics Reliability*.

Conference papers:

1. C.-W. Eng, W.-S. Lau, Y.-Y. Jiang, D. Vigar, K.-C. Tee, L. Chan, V. S.-W. Lim and A. Trigg, “Improving the Accuracy of Modified Shift-and-Ratio Channel Length Extraction Method Using Scanning Capacitance Microscopy,” in *Proc. Int. Conf. Solid State Devices and Materials (SSDM)*, p. 492, 2003.
2. C. W. Eng, W.S. Lau, Y. Y. Jiang, D. Vigar, V. S. W. Lim and A. Trigg, “A Simpler and Convenient Effective Channel Length Extraction Method for Routine Electrical Testing of Deep Sub-quarter-micron MOSFET,” Technical Digest, *Symposium of Microelectronics*, p. P05, 2004.
3. W.S. Lau, K.S. See, C.W. Eng, W. K. Awl, K. H. Jo, K. C. Tee, James Y. M. Lee, Elgin K. B. Quek, H. S. Kim, S. T. H. Chan, L. Chan, “Anomalous narrow wide effect in NMOS and PMOS surface channel transistors using shallow trench isolation,” *Proc. IEEE Conference on Electron Devices and Solid-State Circuits*, p.773, 2005.
4. W.S. Lau, C.W. Eng, K.M. Tee, S.Y. Siah, D. Vigar, Y.T. Kim, Manni Lal, M. Bhat and L. Chan, “A mechanism of increase in the on-current and off-current due to a

slightly smaller spacer in state-of-the art p-channel MOS transistors during manufacturing,” *Proc. IEEE Conference on Electron Devices and Solid-State Circuits*, p.777, 2005.

6.2 Recommendations for future work

6.2.1 Application and improvement of our L_{eff} extraction method

Based on the experimental data we had collected, our methods prove to be reasonably accurate in extracting L_{eff} . However, our method may not be sensitive enough to detect a small change in channel length due to process variation. At this point in time our proposed methods were just a starting point in developing a method more suitable to be use in automatic wafer electrical test systems for routine process monitoring.

MS&R method is recommended to be use for routine process monitoring as it is much simpler to implement then the subthreshold regime based method. However, more data have to be collected to verify the sensitivity of our method to process variation.

To improve the accuracy of the subthreshold regime-based method, we suggest looking into the expression of the subthreshold slope for short channel transistor. In our method, the *sub-threshold swing*, S was defined by

$$S = 2.3 \frac{mkT}{q} \text{ and it was assumed to be equal for both the short and long channel}$$

device.

However, for short channel device, S can be defined as

$$S = 2.3 \frac{mkT}{q} \left(1 + \frac{11t_{ox}}{W_{dm}} e^{-pL/2(W_{dm}+3t_{ox})} \right)$$

The accuracy of the extracted the L_{eff} can be improved with the proper modeling and extraction of the *sub-threshold swing* for short channel device.

6.2.2 Theoretically calculation of remote Coulomb scattering

In Chapter 4, based on our experimental study on the mobility of the short and long channel transistors, we have verified that at lower gate voltage range near to threshold voltage, the mobility of the short transistor is approximately equal to that of the long transistor due to remote Coulomb scattering (RCS). For a more complete verification, a more theoretically vigorous quantum-mechanical treatment of the effect of RCS on the effective carrier mobility of very short channel transistor operating in the sub-threshold region should be attempted.

6.2.3 Extension of our L_{eff} extraction method to latest technology

This new method to measure L_{eff} should be further demonstrated for MOS transistors fabricated by 65 nm CMOS or even newer 45 nm CMOS where the physical gate oxide thickness can be as thin as 1.1 nm. It should also be tested for MOS transistors fabricated by CMOS technology enhanced by stress engineering. It will be interesting to investigate the effect of the various mobility enhancement techniques have on the accuracy of our L_{eff} extraction method. Another challenge in extracting L_{eff} comes from the application of high-k dielectric in 32 nm CMOS technology. The application and verification of our methods for 32 nm CMOS technology may prove to be valuable. Finally the application of DIBL as an effective measure of effective channel length should be further studied.

Bibliography

- [1] G. E. Moore, "Progress in Digital Integrated Circuit," *IEEE IEDM Tech. Dig.*, p. 11, 1975.
- [2] C. Y. Chang and S. M. Sze, *ULSI Technology*, (McGraw-Hill, Singapore, 2001).
- [3] Semiconductor Industry Association, International Technology Roadmap for Semiconductors: 2004 Edition, Austin, TX: International SEMATECH, 2004.
- [4] Y. Taur, D. A. Buchanan, W. Chen, D. J. Frank, K. E. Ismail, S.-H. Lo, G. A. Sai-Halasz, R. G. Viswanathan, H.-J. C. Wann, S. J. Wind and H.-S. Wong, "CMOS Scaling into the Nanometer Regime," *Proc. IEEE*, Vol. 85, No. 4, p.486, April 1997.
- [5] T. Ghani, K. Mistry, P. Packan, S. Thompson, M. Stettler, S. Tyagi and M. Bohr, "Scaling Challenges and Device Design Requirement for High Performance Sub-50nm Gate Length Planer CMOS transistors," *Symp. on VLSI Tech. Dig.*, p.174, 2000.
- [6] S. Asai and Y. Wada, "Technology Challenges for Integration Near and below 0.1 μm ," *Proc. IEEE*, Vol. 85, No. 4, p.505, April 1997.
- [7] P. M. Zeitzoff, "Challenges and Key Potential Technological Innovations For Scaling MOSFETS Through the End of the Roadmap," *Proc. of the 7th International Conference on Solid-State and Integrated Circuits Technology*, Vol. 1, p.20, October 2004.
- [8] E. D. Boskin, C. J. Spanos, and G. J. Korsh, "A Method for Modeling the Manufacturability of IC Designs," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 7, No. 3, p. 298, August 1994.

Bibliography

- [9] M. Orshansky, J. An, C. Jiang, B. Liu, C. Riccobene, and C. Hu, "Efficient Generation of Pre-Silicon MOS Model Parameters for Early Circuit Design," *IEEE Journal of Solid-State Circuits*, Vol. 36, No. 1, p.156, January 2001.
- [10] Y. Taur and T. H. Ning, *Fundamental of Modern VLSI Devices*, (Cambridge University Press, 1998) p. 202.
- [11] S. Ogura, P. J. Tsang, W. W. Walker, D. L. Critchlow, and J. F. Shepard, "Design and Characteristics of the Lightly Doped Drain-Source (LDD) insulated Gate Field-Effect Transistor," *IEEE Trans. Electron Devices*, Vol. ED-27, No. 8, p.1359, August 1980.
- [12] P. E. Cottrell, R. R. Troutman, and T. H. Ning, "Hot-Electron Emission in N-Channel IGFET's," *IEEE Journal of Solid-State Circuits*, Vol. SC-14, No. 2, p.520, April 1979.
- [13] Richard B. Fair, and Robert C. Sun, "Threshold-Voltage Instability in MOSFET's Due to Channel Hot-Hole Emission," *IEEE Trans. Electron Devices*, Vol. ED-28, No. 1, p.83, January 1981.
- [14] J. Ida, A. Kita and F. Ichikawa, "A New Extraction Method for Effective Channel Length on Lightly Doped Drain MOSFET'S," *Proc. IEEE Int. Conference on Microelectronic Test Structures*, Vol. 3, p. 117, March 1990.
- [15] G. Niu, J. D. Cressler, S. J. Mathew, and S. Subbanna, "A Channel Resistance Derivative Method for Effective Channel Length Extraction in LDD MOSFET's," *IEEE Trans. Electron Devices*, Vol. 47, No. 3, March 2000.
- [16] B. J. Sheu, and P. K. Ko, "A Capacitance Method to Determine Channel Lengths for Conventional and LDD MOSFET," *IEEE Electron Dev. Lett.*, Vol. EDL-5, No. 11, p.491, November 1984.

Bibliography

- [17] S. H. Chuang, T. Carns, S. K. Lee, L. DeBurler and J. E. Berg, "Measuring the Effective Channel Length of the Deep Submicron MOSFET and Channel Broadening Effect," *Proc. of the Thirteenth Biennial University/Government/Industry Microelectronics Symposium*, p. 176, 1999.
- [18] T. Hori, "A 0.1- μm CMOS Technology with Tilt-Implanted Punchthrough Stopper (TIPS)," *IEEE IEDM Tech. Dig.*, p. 75, 1994.
- [19] C. F. Codella and S. Ogura, "Halo Doping Effects in Submicron DI-LDD Device Design," *IEEE IEDM Tech. Dig.*, p. 230, 1985.
- [20] A. Hori, H. Nakaoka, H. Umimoto, K. Yamashita, M. Takase, N. Shimizu, B Mizuno and S. Odanaka, "A 0.05 μm -CMOS with Ultra Shallow Source/Drain Junction Fabricated by 5 keV Ion Implant and Rapid Thermal Annealing," *IEEE IEDM Tech. Dig.*, p. 485, 1994.
- [21] B. Yu, C. H. J. Wann, E. D. Nowak, K. Noda and C. Hu, "Short-Channel Effect Improved by Lateral Channel-Engineering in the Deep-Submicronmeter MOSFET's," *IEEE Trans. Electron Devices*, Vol. 44, No. 4, p. 627, April 1997.
- [22] R. Gwoziecki, T. Skotnicki, P. Bouillon and P. Gentil, "Optimization of V_{th} Roll-Off in MOSFET's with Advanced Channel Architecture-Retrograde Doping and Pockets," *IEEE Trans. Electron Devices*, Vol. 46, No. 7, p. 1551, July 1999.
- [23] H. Van Meer, K. Henson, J.-H. Lyu, M. Rosemeulen, S. Kubicek, N. Collaert and K. D. Meyer, "Limitations of Shift-and-Ratio Based L_{eff} Extraction Techniques for MOS Transistors with Halo or Pocket Implants," *IEEE Electron Dev. Lett.*, Vol. 21, No. 3, p.133, March 2000.
- [24] Y. Taur, "MOSFET Channel Length: Extraction and Interpretation," *IEEE Trans. Electron Devices*, Vol. 47, No. 1, p.160, January 2000.

Bibliography

- [25] K. Terada and H. Muta, "A New Method to Determine Effective MOSFET Channel Length," *Jpn. J. Appl. Phys.* Vol. 18, No. 5, p.953 May 1979.
- [26] J. G. J. Chern, P. Chang, R. F. Motta and N. Godinho. "A New Method To Determine MOSFET Channel Length," *IEEE Electron Dev. Lett.*, Vol. EDL-1, No. 9, p.170, September 1980.
- [27] G. J. Hu, C. Chang and Y.-T. Chia, "Gate-Voltage-Dependent Effective Channel Length and Series Resistance of LDD MOSFET's," *IEEE Trans. Electron Devices* Vol. ED-34, No. 12, p.2469, December 1987.
- [28] K. K. Ng and J. R. Brews, "Measuring the Effective Channel length of MOSFETs," *IEEE Circuits and Devices Magazine*, Vol. 6, p.33, 1990.
- [29] Y. Taur, D. S. Zicherman, D. R. Lombardi, P. J. Restle, C. H. Hsu, H. I. Hanafi, M. R. Wordeman, B. Davari and G. G. Shahidi, "A New Shift and Ratio Method for MOSFET Channel-Length Extraction," *IEEE Electron Dev. Lett.*, Vol. 13, p.267, 1992.
- [30] S. Biesemans, M. Hendriks, S. Kubicek, and K. De Meyer, "Accurate Determination of Channel Length, Series Resistance and Junction Doping Profile for MOSFET Optimisation in Deep Submicron Technologies," *VLSI Symp. Technology Dig.*, p.166, 1996.
- [31] Y.-S. Jean and C.-Y. Wu, "A New Extraction Algorithm for the Metallurgical Channel Length of Conventional and LDD MOSFET's," *IEEE Trans. Electron Devices* Vol. 43, No. 6, p.946, June 1996.
- [32] T.-H. Chang, J.-G. Lo, T.-C. Kuo, C. C.-H. Hsu, S.-Y. Yu, K.-F. Tseng and L. S. Lu, "Effective Channel Length and Source-Drain Series-Resistance Determine

Bibliography

- after Electrical Gate Length Verification of Metal-Oxide-Semiconductor Field-Effect-Transistor,” *Jpn. J. Appl. Phys.* Vol. 37, Part 1, No. 3A, p.796 March 1998.
- [33] K. Yamaguchi, H. Amishiro, M. Yamawaki and S. Assai, “A New Variational Method to Determine Effective Channel Length and Series Resistance of MOSFET’s,” *Proc. IEEE 1998 Int. Conference on Microelectronic Test Structures*, Vol. 11, p. 123, March 1998.
- [34] S. H. Chuang, T. Carns, S. K. Lee, L. DeBurler and J. E. Berg, “Measuring the Effective Channel Length of the Deep Submicron MOSFET and Channel Broadening Effect,” *Proc. of the Thirteenth Biennial University/Government/Industry Microelectronics Symposium*, p. 176, 1999.
- [35] Q. Ye, Y. Li, W. Tonti, W. Berry, C. Parks and R. Mohler, “Improving the Accuracy of “Shift and Ratio” Channel Length Extraction Method In Deep Submicron Technology,” *IEEE International Reliability Workshop Final Report* (IEEE, New York) p.102, 2000.
- [36] C. Mourrain, B. Cretu, G. Ghibaudo, P. Cottin, “New Method for Parameter Extraction in Deep Submicrometer MOSFETS,” *Proc. IEEE 1998 Int. Conference on Microelectronic Test Structures*, p.181, 2000.
- [37] B. Cretu, T. Boutchacha, G. Ghibaudo, F. Balestra, “New Ratio Method for Effective Channel Length and Threshold Voltage Extraction in MOS Transistors,” *Electronics Lett.* Vol. 37, No. 11, p.717, May 2001.
- [38] C.-W. Eng , W.-S. Lau , D. Vigar and J. Y.-M. Lee, “An Improved Shift-and-Ratio Effective Channel Length Extraction Method for Metal Oxide Silicon Transistors with Halo/Pocket Implants,” *Jpn. J. Appl. Phys.* Vol. 42, Part 1, No. 5A, p.2621, May 2003.

Bibliography

- [39] C.-W. Eng, W.-S. Lau, Y.-Y. Jiang, D. Vigar, K.-C. Tee, L. Chan, V. S.-W. Lim and A. Trigg, "Improving the Accuracy of Modified Shift-and-Ratio Channel Length Extraction Method Using Scanning Capacitance Microscopy," *Jpn. J. Appl. Phys.*, Vol. 43, No. 4B, p.1869, 2004.
- [40] M. Fathipour, E. Fathi, B. Afzal and A. Khakifirooz, "An improved Shift-and-Ratio L_{eff} Extraction Method for MOS Transistors with halo-pocket implants," *Solid-State Electronics*, Vol. 48, p.1829, 2004.
- [41] J.-C. Guo, C. C.-H. Hsu, P.-S. Lin and S.S. Chung, "An Accurate "Decoupled C-V" Method for Characterizing Channel and Overlap Capacitance of Miniaturized MOSFET," International Symposium on VLSI Technology, Systems, and Applications, p.256, 1993.
- [42] J.-C. Guo, S. S.-S. Chung and C. C.-H. Hsu, "A New Approach to Determine the Effective Channel Length and the Drain-and-Source Series Resistance of Miniaturized MOSFET's," *IEEE Trans. Electron Devices*, Vol. 41, No. 10, p.1811 October 1994.
- [43] C.-L. Huang, J. V. Faricelli, D. A. Antoniadis, N. A. Khalil and R. A. Rios, "An Accurate Gate Length Extraction Method for Sub-Quarter Micron MOSFET's," *IEEE Trans. Electron Devices*, Vol. 43, No. 6, p. 958 June 1996.
- [44] Z. Latif, A. O.-Conde, J. J. Liou and F. J. G. Sanchez, "On the Extraction of Effective Channel Length of MOSFETs," *Proc. 21st International Conference on Microelectronics*, Vol. 1, p.281, September 1997.
- [45] S. Lee, "A New RF Capacitance Method to Extract the Effective Channel Length of MOSFET's Using S-Parameters," *Proc. IEEE Hong Kong Electron Device Meeting*, p.56, 2000.

Bibliography

- [46] T.S. Hsieh, Y.W. Chang, W.J. Tsai, T.C. Lu, "A New Extraction Approach for Device with Pocket Implants," *Proc. Int. Conference on Microelectronic Test Structures*, Vol. 14, p. 15, March 2001.
- [47] H.-S. Huang, J.-S. Shiu, S.-J. Lin, J.-W. Chou, R. Lee, C. Chen and G. Hong, "A Modified Capacitance-Voltage Method Used for L_{eff} Extraction and Process Monitoring in Advance 0.15 μm Complementary Metal-Oxide-Semiconductor Technology and Beyond," *Jpn. J. Appl. Phys.*, Vol. 40, No. 3A, p.1222, March 2001.
- [48] S. Lee and S. Youn, "Improved Extraction Method for Effective Channel Length of Deep-Submicrometer MOSFETs," *Electronics Lett.*, Vol. 40, No. 4, p. 274, February 2004.
- [49] Q. Ye and S. Biesemans, " L_{eff} Extraction for sub-100 nm MOSFET Devices," *Solid State Electronics*, Vol. 48, p.163, 2004.
- [50] K. Iniewski and C. A. T. Salama, "New Method for Extraction of Effective Channel Length in Submicron MOSFETs," *Electronics Lett.*, Vol. 27, No.6, p. 508, March 1991.
- [51] X. Zhou, K. Y. Lim and D. Lim, "A New "Critical-Current at Linear-Threshold" Method for Direct Extraction of Deep-Submicron MOSFET Effective Channel Length," *IEEE Trans. Electron Devices*, Vol. 46, No. 7, p. 1492 July 1999.
- [52] S. Hong, Y. Zhang, Y. Luo, T. Suligoj, S.-D. Kim, J. C.S. Woo, R. Li, B.-W. Min, B. Hradsky, A. Vandooren, B.-Y. Nguyen, and K. L. Wang, "Novel Direct-Tunneling-Current (DTC) Method for Channel Length Extraction Beyond Sub-50nm Gate CMOS," *IEDM Tech. Dig.*, p.297, 2001.

Bibliography

- [53] M. Marin, M. J. Deen, M. de Murcia, P. Llinares and J.-C. Vildeuil, "A New Method for the Channel-Length Extraction in MOSFETs with Sub-2-nm Gate Oxide," *IEEE Electron Dev. Lett.*, Vol. 25, No. 4, p.202, April 2004.
- [54] Y. Taur, Y.-J. Mii, R. Logan and H.-S. Wong, "On Effective Channel Length in 0.1- μm MOSFET's," *IEEE Electron Dev. Lett.*, Vol. 16, No. 4, p.136, April 1995.
- [55] K. Ahmed, I. De, C. Osburn, J. Wortman and J. Hauser, "Limitations of the Modified Shift-and-Ratio Technique for Extraction of the Bias Dependence of L_{eff} and R_{sd} of LDD MOSFET's," *IEEE Trans. Electron Devices* Vol. 47, No. 4, p.891, April 2000.
- [56] T. Hori, "A 0.1- μm CMOS Technology with Tilt-Implanted Punchthrough Stopper (TIPS)," *IEDM Tech. Dig.*, p. 75, 1994.
- [57] B. Yu, C. H. J. Wann, E. D. Nowak, K. Noda and C. Hu, "Short-Channel Effect Improved by Lateral Channel-Engineering in the Deep-Submicrometer MOSFET's," *IEEE Trans. Electron Devices*, Vol. 44, No. 4, p. 627, April 1997.
- [58] R. Gwoziecki, T. Skotnicki, P. Bouillon and P. Gentil, "Optimization of V_{th} Roll-Off in MOSFET's with Advanced Channel Architecture-Retrograde Doping and Pockets," *IEEE Trans. Electron Devices*, Vol. 46, No. 7, p. 1551, July 1999.
- [59] D. Esseni, H. Iwai, M. Saito and B. Ricco, "Nonscaling of MOSFET's Linear Resistance in Deep Submicrometer Regime," *IEEE Electron Dev. Lett.*, Vol. 19, No. 4, p.131, April 1998.
- [60] N. D. Arora, D. A. Bell and L. A. Bair, "An accurate Method of determining MOSFET Gate Overlap Capacitance," *Solid State Electronics*, Vol. 35, No. 12, p.1817, 1992.

Bibliography

- [61] C. C. McAndrew, G. Zaneski, P. A. L. and S. G. Ayyar, "Accurate Characterization of MOSFET Overlap/Fringing Capacitance for Circuit Design," *Proc. Int. Conference on Microelectronic Test Structures*, Vol. 7, p. 15, March 1994.
- [62] F. Pregalding, C. Lallement and D. Mathiot, "A Simple Efficient Model of Parasitic Capacitances of Deep-Submicron LDD MOSFETs," *Solid State Electronics*, Vol. 46, p.2191, 2002.
- [63] Y. K and H. C., "MOS Capacitance Measurements for High Leakage Thin Dielectrics," *IEEE Trans Electron Dev.*, Vol. ED-46 p.1500, 1999.
- [64] Z.-H. Liu, C. Hu, J.-H. Huang, T.-Y. Chan, M.-C Jeng., P. K Ko. And Y. C. Cheng, "Threshold Voltage Model for Deep-Sub-Micrometer MOSFET's," *IEEE Trans. Electron Dev.*, Vol. ED-40, p.86, 1993.
- [65] R. van Langevelde, A.J. Scholten, R. Duffy_, F.N. Cubaynes_, M.J. Knitel and D.B.M. Klaassen, "Gate current: Modeling, **DL** extraction and impact on RF performance," *IEDM Tech. Dig.*, p.289, 2001.
- [66] N. Yang, W. K. Henson and J. J. Wortman, "Analysis of Tunneling Currents and Reliability of NMOSFET's with Sub-2 m Gate Oxide," *IEDM Tech. Dig.*, p.453, 1999.
- [67] Dieter K. Schroder, *Semiconductor Material and Device Characterization*, (A Wiley-Interscience Publication, second edition), p. 649.
- [68] R. N. Kleiman, M. L. O'Malley, F. H. Baumann, J. P. Garno, W. G. Timp, and G. L. Timp, "Direct Channel Length Determination of sub-100nm MOS Devices Using Scanning Capacitance Micropy," *Symposium on VLSI Technology Digest of Technical Papers*, p.138, 1998.

Bibliography

- [69] W. Timp, M. L. O'Malley, R. N. Kleiman, J. P. Garino, "Two-Dimensional Dopant Profiling of a 60nm Gate Length nMOSFET Using Scanning Capacitance Microscopy," *IEDM Tech. Dig.*, p. 555, 1998.
- [70] P. A. Rosenthal, Y. Taur and E. T. Yu, "Direct Measurement and Characterization of n^+ Superhalo Implants in a 120 nm Gate-Length Si Metal-Oxide-Semiconductor Field-Effect Transistor Using Cross-Sectional Scanning Capacitance Microscopy," *Appl. Phys. Lett.*, Vol. 81, p. 3993, 2002.
- [71] Y. Y. Jiang, "Scanning Capacitance Microscopy and Related Technologies For Semiconductor Device Characterizations," Master of Engineering thesis, Nanyang Technological University, 2003.
- [72] David B. Williams and C. Barry Carter, *Transmission Electron Microscopy*, (Plenum Press).
- [73] S. C. Sun and J. D. Plummer, "Electron Mobility in Inversion and Accumulation Layers on Thermally Oxidized Silicon Surface," *IEEE. Journal of Solid-State Circuits*, Vol. SC-15, No. 4, p.562, 1980.
- [74] J. R. Hauser, "Extraction of Experimental Mobility Data for MOS Devices," *IEEE Trans Electron Dev.*, Vol. 43, No. 11, p.1981, 1996.
- [75] G. Niu, J. D. Cressler, S. J. Mathew and S. Subbanna, "A total resistance Slope-Based Effective Channel Mobility Extraction Method for Deep submicrometer CMOS Technology," *Solid State Electron.*, Vol. 44, p. 1187, 1999.
- [76] C.-L. Huang, J. V. Faricelli and N. D. Arora, "A New Technique for Measuring MOSFET Inversion Layer Mobility," *IEEE Trans Electron Dev.*, Vol. 40, No. 6, p.1134, 1993.

Bibliography

- [77] B Cheng and J. Woo, "A Temperature-Dependent MOSFET Inversion Layer Carrier Mobility Model for Device and Circuit Simulation," *IEEE Trans Electron Dev.*, Vol. 44, No. 2, p.343, 1997.
- [78] S.-I. Takagi, A. Toriumi, M. Iwase and H. Tango, "On the Universality of Inversion Layer Mobility in Si MOSFET's: Part II- Effects of Surface Orientation," *IEEE Trans Electron Dev.*, Vol. 41, No. 12, p.2363, 1994.
- [79] S.-I. Takagi, A. Toriumi, M. Iwase and H. Tango, "On the Universality of Inversion Layer Mobility in Si MOSFET's: Part I- Effects of Substrate Impurity Concentration," *IEEE Trans Electron Dev.*, Vol. 41, No. 12, p.2357, 1994.
- [80] V. M. Agostinelli, H. Shin and A. F. Tasch, "A Comprehensive Model for Inversion Layer Hole Mobility for Simulation of Submicrometer MOSFET's," *IEEE Trans. Electron Dev*, Vol. 38, p.151, 1991.
- [81] H. Shin, G. M. Yeric, A. F. Tasch and C. M. Maziar, "Physically-Based Models for Effective Mobility and Local-Field Mobility of Electrons in MOS Inversion Layers," *Solid State Electron*, Vol. 34, p. 545, 1991.
- [82] J. Li and T.-P. Ma, "Scattering of Silicon Inversion Layer Electron by Metal/Oxides Interface Roughness," *J. Appl. Phys.*, Vol. 62, No. 10, p. 4212, 1987.
- [83] S.-I. Saito, K. Torii, Y. Shimamoto, S. Tsujikawa, H. Hamamura, O. Tonomura, T. Mine, D. Hisamoto, T. Onai, J. Yugami, M. Hiratani and S. Kimura, "Effects of Remote-Surface-Roughness Scattering on Carrier Mobility in Field-Effect-Transistors with Ultrathin Gate Dielectrics," *J. Appl. Phys.*, Vol. 84, No. 8, p. 1395, 2004.

Bibliography

- [84] M. S. Krishnan, L. Chang, T.-J. King, J. Bokor and C. Hu, "MOSFETs with 9 to 13 Å Thick Gate Oxides," *IEDM*, p.241, 1999.
- [85] N. Yang, W. K. Henson, J. R. Hauser and J. J. Wortman, "Estimation of the Effects of Remote Charge Scattering on Electron Mobility of n-MOSFET's with Ultrathin Gate Oxides," *IEEE Trans Electron Dev.*, Vol. 47, No. 2, p.440, 2000.
- [86] M. S. Krishnan, Y. C. Yeo, Q. Lu, T.-J. King, J. Bokor and C. Hu, "Remote Charge Scattering in MOSFETs with Ultra-Thin Gate Dielectrics," *IEDM*, p.571, 1998.
- [87] S.-I. Saito, K. Torii, M. Hiratani and T. Onai, "Improved Theory for Remote-Charge-Scattering-Limited Mobility in Metal-Oxide-Semiconductor Transistors," *Appl. Phys. Lett.*, Vol. 81, No. 13, p.2391, 2002.
- [88] S.-I. Takagi and M. Takayanagi, "Experimental Evidence of Inversion-Layer Mobility Lowering in Ultrathin Gate Oxide Metal-Oxide-Semiconductor Field-Effect-Transistors with Direct Tunneling Current," *Jpn. J. Appl. Phys.* Vol. 41, p. 2348, 2002.
- [89] F. Gamiz, A. Godoy, and J. B. Roldan, "A New Remote Coulomb Scattering Model for Ultrathin Oxide MOSFETs," *International Conference on Simulation of Semiconductor Process & Devices*, p.47, 2003.
- [90] D. Esseni and A. Abramo, "Modeling of Electron Mobility Degradation by Remote Coulomb Scattering in Ultrathin Oxide MOSFETs," *IEEE Trans Electron Dev.*, Vol. 50, No.7, p.1665, 2003.
- [91] T. Ishihara, J. Koga, S. Takagi and K. Matsuzawa, "Screening Effect on Remote Coulomb Scattering Due to Impurities in Polysilicon Gate MOSFET," *Extended*

Bibliography

- Abstracts of the 2003 International Conference on Solid State Devices and Materials*, p. 12, 2003.
- [92] F. Gamiz and M. V. Fischetti, "Remote Coulomb Scattering in Metal-Oxide-Semiconductor Field Effect Transistors: Screening by Electrons in the Gate," *Appl. Phys. Lett.*, Vol. 83, No. 23, p.4848, 2003.
- [93] Taurus-TSUPREM4 is a trademark of Synopsys Inc., 2006.
- [94] G. Querirolo, E. Servida, L. Baldi, G. Pignatelli, A. Armigliato, S. Frabboni and F. Corticelli, "Dopant Activation, Carrier Mobility, and TEM Studies in Polycrystalline Silicon Films," *J. Electrochem. Soc.* Vol. 137 p.967, 1990.
- [95] K. Romanjek, F. Andrieu, T. Ernst and G. Ghibaudo, "Improved Split C-V Method for Effective Mobility Extraction in sub-0.1- μm Si MOSFETs", *IEEE Electron Dev. Lett.*, Vol. 25, No. 8, p.583, August 2004.
- [96] W. G. En, D.-H. Ju, D. Chan, S. Chan and O. Karlsson, "Reduction of STI Active Stress on 0.18 μm SOI Devices through Modification of STI Process," *IEEE International SOI Conference*, p. 87, 2001.
- [97] G. Scott, J. Lutze, M. Rubin, F. Nouri and M. Manley, "NMOS Drive Current Reduction Caused by Transistor Layout and Trench Isolation Induced Stress," *IEDM*, p. 827, 1999.
- [98] F. Nouri, O. Laparra, H. Sur, G. C. Tai, D. Pramanik and M. Manley, "A Manufacturable Shallow Trench Isolation Process for 0.18 μm and beyond Optimization, Stress Reduction and Electrical Performance," *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, p. 413, 1998.

Bibliography

- [99] M. Miyamoto, H. Ohta, Y. Kumagai, Y. Sonobe, K. Ishibashi, and Y. Tainaka, "Impact of Reducing STI-Induced Stress on Layout Dependence of MOSFET Characteristics," *IEEE Trans. Electron Dev.*, Vol. 51, No. 3, p. 440, March 2004.
- [100] J. J. Liou, A. O. -C and F. G. Sanchez, "Extraction of the Threshold Voltage of MOSFETs: an Overview," *Proc. IEEE Electron Device Meeting*, Hong Kong, p. 31. 1997.
- [101] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*, (Cambridge Univ. Press, New York, 1998), p. 125.
- [102] K. Terada, K. Nishiyama and K. Hatanaka, "Comparison of MOSFET-Threshold-Voltage Extraction Methods," *Solid-State Electron*, Vol. 45, p.35, 2001.
- [103] H.-S. Wong, M.H. White, T.J. Krutsick and R.V. Booth, "Modeling of Transconductance Degradation and Extraction of Threshold Voltage in Thin Oxide MOSFET's," *Solid-State Electron*, Vol. 30, p. 953, 1987.
- [104] B. El-Kareh, W.R. Tonti and S.L. Titicomb, "A submicron MOSFET Parameter Extraction Technique," *IBM J. Res. Develop.*, Vol. 34, p. 243, 1990.
- [105] K. Aoyama, "A Method for Extracting the Threshold Voltage of MOSFET's Based on Current Components," *Simulation of Semiconductor Devices and Processes*, Vol. 6, p. 118, 1995.
- [106] M. Tsuno, M. Suga, M. Tanaka, K. Shibahara, M. Miura-Mattausch and M. Hirose, "Physically-Based Threshold Voltage Determination for MOSFET's of All Gate Lengths," *IEEE Trans. Electron Dev.*, Vol. 46, No. 7, p.1429, July1999.
- [107] F.J. Garcia Sanchez, A. Ortiz-Conde, G. De Mercato, J.A. Salcedo, J.J. Liou and Y. Yue, "New Simple Procedure to Determine the Threshold Voltage of MOSFETs," *Solid-State Electron.*, Vol. 44, p.673, 2000.

Bibliography

- [108] J. He, X. Xi, M. Chan, K. Cao, C. Hu, Y. Li, X. Zhang, R. Huang and Y. Wang, "Normalized Mutual Integral Difference Method to Extract threshold Voltage of MOSFETs," *IEEE Electron Dev. Lett.*, Vol. 23, p. 428, July 2002.
- [109] W.S. Lau, Peizhen Yang, C.W. Eng, V. Ho, C.H. Loh, S.Y. Siah, D. Vigar and L. Chan, "A study of the linearity between I_{on} and $\log I_{off}$ of modern MOS transistors and its application to stress engineering," accepted for publication in *Microelectronics Reliability*.
- [110] J. S. Wong, J. G. Ma, K. S. Yeo and M. A. Do, "A New Approach for the Extraction of Threshold Voltage for MOSFET's," *International Conference on Modeling and Simulation of Microsystems*, p. 534, 2001.
- [111] W. Y. Choi, D.-S. Woo, B. Y. Choi, J. D. Lee and B.-G. Park, "Stable Extraction of Threshold Voltage Using Transconductance Change Method for CMOS Modeling, Simulation and Characterization," *Jpn. J. Appl. Phys.* Vol. 43, No. 4B, p.1759, May 2004.
- [112] A. Ortiz-Conde, E. D. G. Fernandes, J. J. Liou, M. R. Hassan, F. J. Garcia-Sanchez, G. D. Mercato, and W. Wong, "A New Approach to Extract the Threshold Voltage of MOSFET's," *IEEE Trans. Electron Dev.*, Vol. 44, No. 9, p.1523, 1997.
- [113] R. Narayana, Z. Latif, A. Ortiz-Conde, J. J. Liou, L. Golovanova, W. Wong and F. J. Garcia Sanchez, "On the Reverse Short-Channel Effects of Submicron MOSFETs," *Conference Record Southcon*, p. 345, June 1996.
- [114] C.-Y. Lu and J. M. Sung, "Reverse Short-Channel Effect on Threshold Voltage in Submicrometer Salicide Devices," *IEEE Electron Dev. Lett*, Vol.10, p. 446, October 1989.

Bibliography

- [115] C.-Y. Chang, C.-Y. Lin, J. W. Chou, C. C.-H. Hsu, H.-T. Pan, and J. Ko, “Anomalous Reverse Short Channel Effect in P⁺ Polysilicon Gated P-channel MOSFET,” *IEEE Electron Dev. Lett*, Vol.15, p. 437, November 1994.
- [116] J. F. Buller, J. Cheek, D. Wristers, D. Kadosh and M. Duane, “Improved p-ch MOS Transistor Performance with an Arsenic Super Steep Retrograde Channel Profile,” *SPIE Conference on Microelectronic Device Technology III*, Vol.3881 p.138, 1999.
- [117] A. Lochtefeld, I. J. Djomehri, G. Samudra and D. A. Antoniadis, “New Insights into Carrier Transport in n-MOSFETs,” *IBM J. Res & Dev.*, Vol. 46, p. 347, March/May 2002.

Appendix A

The front end process

Appendix A

NITRIDE REMOVAL, SACRIFICIAL OXIDE GROWTH

N-WELL, PUNCHTHROUGH IMPLANT, THRESHOLD ADJUSTMENT IMPLANT

P-WELL, PUNCHTHROUGH IMPLANT THRESHOLD ADJUSTMENT IMPLANT

Appendix A

GATE OXIDE GROWTH, POLY DEPOSITION

PHOTORESIST DEPOSITION

PHOTORESIST ETCH

Develop Image Critical Dimension (DICD) is an in-line monitoring parameter. It's a measurement of the dimension of the photo resist after it was etched. This measurement was done by the CD-SEM (Critical Dimension Scanning Electron Microscope) tool while the wafer is still in fabrication in the fab.

Appendix A

POLY ETCH

Final Image Critical Dimension (FICD) is an in-line monitoring parameter. It's a measurement of the dimension of the poly after it was etched. This measurement was done by the CD-SEM (Critical Dimension Scanning Electron Microscope) tool while the wafer is still in fabrication in the fab.

HALO IMPLANT

Appendix A

SOURCE DRAIN EXTENSION

L SHAPE NITRIDE SPACER

Appendix A

DEEP SOURCE DRAIN IMPLANT

COBALT SALICIDE

Appendix B

B1: Threshold Voltage

The threshold voltage, V_{th} is an important MOSFET parameter which directly determines the turn-on and turn-off of the MOS transistors. Therefore, its accurate estimation is a key concern for device characterization and modeling. However, V_{th} is a voltage that is not uniquely defined. Various definitions exists, a commonly used definition of V_{th} is the gate voltage when the surface potential or band bending reaches $2\phi_B$ and the silicon charge is equal to the bulk depletion charge for that potential. The threshold voltage for large-geometry, n-channel MOSFET on uniformly doped substrates with no short- or narrow-channel effect is given by

$$V_{th} = V_{fb} + 2\phi_B + \frac{\sqrt{4\epsilon_{si}qN_a\phi_B}}{C_{ox}} \quad (B1)$$

where V_{fb} is the flatband voltage, C_{ox} is the gate oxide capacitance per unit area, ϵ_{si} is the silicon permittivity, q is the electronic charge, N_a is the acceptor doping concentration, ϕ_B is the difference between the Fermi level in the channel region and the intrinsic level.

The extraction technique of V_{th} starts with the drain current measured as a function of gate voltage at low drain voltage of typically 50~100 mV to ensure operation in the linear MOSFET region. V_{th} extracted in the linear region is defined as linear V_{th} . The threshold voltage can also be determined in the MOSFET saturation regime. Setting $V_{gs}=V_{ds}$ ensures operation in the saturation region, V_{th} extracted at saturation region is defined as saturation V_{th} .

Appendix B

There are a number of threshold voltage extraction methods, however, the extracted values for the V_{th} vary from method to method [90, 92]. The constant-current (CC) method is widely used in the industry because of its simplicity, the threshold voltage is defined as the V_{gs} when $I_{ds} = I_{do} * (W_{eff}/L_{eff})$. However, it has the disadvantage of being strongly dependent of the arbitrary choice of current I_{do} as shown in Fig. B1. [90]

Fig. B1: Qualitative plot to illustrate the different threshold voltage determined from different drain current. [90]

The linear-extrapolation (LE) method is the most extensively employed by researchers, but it presents the disadvantage of being dependent of the series resistances of the MOSFET [90]. The LE method determines V_{th} by extrapolating linearly plotted $I_{ds}(V_{gs})$ to zero. This method is valid only for low drain voltages (e.g., below 0.1 V), since I_{ds} deviates from linearity as a function of V_{gs} for increased V_{ds} values. Therefore the SRE method, which determines V_{th} by a square root extrapolation of I_{ds} is introduced for high V_{ds} . In addition to a mismatch between two methods, the extracted values do not meet the

Appendix B

threshold condition due to the assumption that the carrier mobility is always independent of V_{gs} .

Transconductance -linear-extrapolation (GMLE) method, this method extracts V_{th} by the linear extrapolation of g_m to zero. The GMLE method utilizes the linear dependence of μ_{eff} as a function of $(V_{gs}-V_{th})$, which originates from the Coulomb scattering. Fig. B2 illustrates the extraction procedure of a fabricated MOSFET at $V_{ds}=0.05V$ as an example.

Fig. B2. Extraction procedure by the GMLE method for fabricated MOSFET with 0.05 μm gate length at $V_{ds} = 0.05 V$. [96]

In the second difference of logarithm of the drain current (SDL) method [95], the gate voltage for the minimum in the $\frac{d^2(I_{ds})}{dV_{gs}^2} - V_{gs}$ curve is defined as the threshold voltage. This corresponds to the condition at which the drift contribution is equal to the diffusion one. The similar condition is given for the transconductance change (TC)

Appendix B

method, which determines at the maximum of derivative of g_m . In the (TC) method [93, 99], the gate voltage at the maximum point on the $\partial^2(I_{ds})/\partial V_{gs}^2 - V_{gs}$ curve is defined as the threshold voltage. These two methods define the threshold voltage as a point of inflection on the $I_{ds}-V_{gs}$ curve between the subthreshold region and the ON region. The SDL and TC method developed to avoid the dependence on the series resistances, is highly sensitive to noise in the measurements because the second derivative is equivalent to a high pass filter.

Fig. B3. Optimized $g_{m2}=dg_m/dV_{GS}$ profiles calculated from the algorithm for optimized node interval. The simulation was done for a 1.5 μm nMOSFET with 25nm gate oxide at 0.1V drain bias. $V_{th}(P)$, $V_{th}(LE)$ and $V_{th}(TC)$ represent the threshold voltage of the classical definition ($\phi_s = 2\phi_f + V_{sb}$), the linear extrapolation and the transconductance change method, respectively. [100]

Appendix B

Another V_{th} extraction method, which was developed based on an integral function was reported in 1997 [101]. This new function, which contains only integration of current and voltage, acts as a low-pass filter. It eliminates the effects of the MOSFET series resistances and therefore provides a more accurate determination of the threshold voltage.

Fig. B4. (a) Drain current versus gate voltage characteristics and (b) corresponding D_{MOS} function obtained from measurements. An increment of 100 mV for the gate bias and $V_d = 100$ mV are used. [101]

Appendix B

For the case of a MOSFET biased in the linear region, the drain current can be expressed as

$$I_D \cong f(V_{GS} - V_T)V_{DS} \quad (B2)$$

where f is a function defined by a particular MOSFET model, and V_{DS} and V_{GS} are the intrinsic gate-source and drain-source voltages, respectively. The two voltages can be related to the external voltages as

$$V_{GS} = V_g - I_D R_s \quad (B3)$$

and

$$V_{DS} = V_d - I_D (R_s + R_d) \quad (B4)$$

Here, V_g and V_d are the extrinsic gate-source and drain-source voltages, respectively, and R_s and R_d are the source and drain series resistances, respectively, which are assumed to be independent of the gate voltage.

The total measured resistance R_m from drain to source is

$$R_m \equiv \frac{V_d}{I_D} = R_d + R_{ch} + R_s \quad (B5)$$

where R_{ch} is the channel resistance, which can be obtained from (B2)

$$R_{ch} \equiv \frac{V_{DS}}{I_D} = \frac{1}{f(V_{GS} - V_T)} \quad (B6)$$

For the strong inversion, the extrinsic gate voltage is high (i.e., $V_{GS} \gg V_T$). Since the extrinsic drain-source voltage is small in the linear region under study, it can be approximated that $(V_{GS} - V_T) \gg (V_g - V_T)$. On the other hand, for weak inversion, $(V_{GS} < V_T)$, the drain current is very small, and the voltage drops in R_s and R_d are much smaller than

Appendix B

V_{GS} . As a result, we can again approximate $(V_{GS} - V_T) \gg (V_g - V_T)$. Thus for either strong or weak inversion, (B6) can be simplified to

$$R_{ch} \approx \frac{1}{f(V_g - V_T)} \quad (B7)$$

Next, a change of variable from V_g to V_{gb} was introduced, it is defined as

$$V_{gb} = V_{max} - V_g \quad (B8)$$

where V_{max} is a constant parameter, which physically represents the maximum gate voltage under consideration. For a MOSFET, R_s and R_d are assumed constant with respect to I_D , but they depend linearly on the variable $(R_m V_{gb})$, therefore, D_{MOS} can be expressed in terms of V_{gb} and $(R_m V_{gb})$ as

$$\begin{aligned} D_{MOS}(V_{gb}, R_m V_{gb}) \\ = \int_0^{(R_m V_{gb})} V_{gb} d(R_m V_{gb}) - \int_0^{V_{gb}} (R_m V_{gb}) dV_{gb} \end{aligned} \quad (B9)$$

Using the method of integration by parts, we have

$$D_{MOS}(V_{gb}, R_m V_{gb}) = R_m V_{gb}^2 - 2 \int_0^{V_{gb}} (R_m V_{gb}) dV_{gb} \quad (B10)$$

and R_m that is a measurable parameter, and D_{MOS} independent of R_s or R_d

The effective electron mobility m_{eff} is a function of V_g

$$m_{eff} = \frac{m_0}{1 + \theta(V_g - V_T)} \quad (B11)$$

where m_0 is the electron mobility in the bulk and the parameter θ accounts for the mobility degradation at the oxides-semiconductor interface. Combining (B2) and (B11) yields

Appendix B

$$I_D = \frac{K(V_g - V_T)}{1 + q(V_g - V_T)} V_{DS} \quad (B12)$$

where K is related to the MOSFET width, MOSFET length, m , and oxide capacitance.

Putting (B12) into (B10), and after some algebraic manipulations,

$$\begin{aligned} D_{MOS}(V_{gb}, R_m V_{gb}) &= \frac{2V_{gb}}{K} + \frac{V_{gb}^2}{K(V_{max} - V_{gb} - V_T)} \\ &+ \frac{2(V_{max} - V_T)}{K} \ln \left[1 - \frac{V_{gb}}{(V_{max} - V_T)} \right] \end{aligned} \quad (B13)$$

The value of D_{MOS} as a function of V_{gb} can be found by substituting the measured or simulated current-voltage data into (B10). This, together with the information of K and V_{max} , allows one to determine V_T from (B13).

Another alternative technique, Max-G-Function, was proposed to extract the threshold voltage from the subthreshold-to-strong inversion transition region of MOSFETs [97]. It uses an auxiliary operator that involves integration of the drain current as a function of gate voltage. This method does not utilize any differentiation of the data thus greatly reduces the effect of possible noise or experimental error.

In order to extract the threshold voltage the drain current is measured versus gate voltage below and above threshold with zero body bias and a small constant value of drain voltage. Next, the following function is numerically calculated from the measured data:

Appendix B

$$G(V_{GS}, I_D) = V_{GS} - 2 \frac{\int_{V_{GSb}}^{V_{GSa}} I_D(V_{GS}) dV_{GS}}{I_D} \quad (\text{B14})$$

where V_{GSb} and V_{GSa} are the lower and upper limits of integration corresponding to gate-to-source voltages below and above threshold, respectively. Function $G(V_{GS}, I_D)$ is linear when plotted as a function of $\ln I_D$ wherever $I_D(V_{GS})$ is exponential, and has the property of vanishing wherever $I_D(V_{GS})$ is linear. Therefore, a plot of G versus $\ln I_D$ should be a straight line below threshold, where the current is dominated by diffusion and consequently it is predominantly exponential. Furthermore, G should drop abruptly to zero as soon as the threshold voltage is surpassed, since above this point the current is dominated by drift and hence it is predominantly lineal.

Fig. B5: Function G for the I_D - V_{GS} characteristics of a MOS transistor. The maximum G represents the value of $V_{th} = 0.85\text{V}$. [97]

Fig.B5 presents such a plot of G which was numerically calculated for a MOS transistor using Eq. (B14). The curve is seen to behave approximately as a straight line

Appendix B

until it reaches a maximum value of about 0.850 V, at which point it falls towards zero. This maximum value of G corresponds to the threshold voltage of the device.

The MP method extracts the values as the point [94], where deviates from an exponential relationship. These values meet the threshold condition from the physical point of view, namely the onset of the inversion-channel creation. V_T is determined from subthreshold slope measurements independent of tolerances in channel dimensions and extrinsic resistance. However, for devices with sub-0.1 μm gate length the high lateral electric field emphasizes the drift contribution even in the subthreshold region, which makes the MP method unreliable.

Fig. B6 Definition of V_T at match point where the measured drain current deviates by 5% from the exponential subthreshold extrapolation. [94]

Appendix B

A new approach for the extraction of threshold voltage (V_{th}) is proposed, namely, the “Third Derivative of Drain-Source Current” method or simply “TD” method [99]. This method extracts the V_{th} by finding the V_{gs} where the third derivative of I_{ds} is maximal. It meets the threshold condition requirement, which is the onset of the inversion channel creation. For this proposed method, V_{th} is define as

$$V_{MO} = V_{FB} + 2f_F + g\sqrt{2f_F - V_{bs}} \quad (B15)$$

where V_{MO} is equal to the V_{gs} where the third derivative of I_{ds} is maximum.

The graphical representation of the proposed method is illustrated in Fig. B7. The transconductance (g_m) curve is obtained by differentiating the drain current (I_{ds}) with respect to V_{gs} and once again to give the curve g_m' whereas the g_m'' curve can be obtained by differentiating the G_m' curve again. The position of the peak of the g_m'' curve on the V_{gs} axis is taken as the threshold voltage.

Fig.B7: Threshold voltage extraction by TD method and TC method. [99]

Appendix B

A novel normalized mutual integral difference (NMID) method had been reported to extract the threshold voltage of MOSFETs. [98] The basic principle of this method is to utilize the exponential-linear characteristics of MOSFETs current so as to obtain the normalized mutual integral difference extreme spectral characteristics.

The drain current of MOSFETs consists of the diffusion and drift components around the threshold voltage V_{th} region. Which component dominates depends on the gate voltage. When the gate voltage is less than V_{th} , the diffusion component dominates and rises exponentially with the gate voltage. Once the gate voltage is above , the drift current starts to dominate and increases approximately linearly with the gate voltage. In the transition region both mechanisms are present and I - V curve behaves as exponent-linear characteristics.

The normalized mutual integral difference method utilizes the exponential-linear characteristic of the drain current versus gate voltage to obtain the threshold voltage. Supposing that the drain current versus gate voltage is a function or $y=f(x)$ or $I_{ds}=f(V_{gs})$, where the gate voltage is the variable and the drain voltage constant. The normalized mutual integral difference operator is defined as

$$P(x) = \frac{\int xdf(x) - \int f(x)dx}{xf(x)} \quad (B16)$$

The function P has no dimension but with the normalized power spectrum form.

Fig.B8 shows the plot of P versus V_{gs} , which was numerically calculated from the measured device characteristics. It is found that P increases exponentially when plotted as a function of V_{gs} as it is less than the threshold voltage, where the current is dominated by the diffusion mechanism. Furthermore, P decreases exponentially as soon as the threshold

Appendix B

voltage is surpassed, since above this point the current is dominated by drift component. The curve peak point can be determined as V_{th} . One of the advantages of this technique is its insensitivity to the parasitic source and drain resistance while sensitivity to the channel length variation is well preserved, both are more important in MOSFET parameter extraction.

Fig. B8. The n -MOSFET characteristics of (a) I - V curve with $V_{ds} = 0.1$ V and (b) the normalized mutual integral difference operator function P versus gate voltage obtained from (a).

Appendix B

B2: Matlab Program source code for threshold voltage calculation

Constant Current Method:

```
function varargout = CC_method_adv_GUI(varargin)
% CC_METHOD_ADV_GUI M-file for CC_method_adv_GUI.fig
%   CC_METHOD_ADV_GUI, by itself, creates a new CC_METHOD_ADV_GUI or raises the
%   existing
%   singleton*.
%
%   H = CC_METHOD_ADV_GUI returns the handle to a new CC_METHOD_ADV_GUI or the
%   handle to
%   the existing singleton*.
%
%   CC_METHOD_ADV_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in CC_METHOD_ADV_GUI.M with the given input arguments.
%
%   CC_METHOD_ADV_GUI('Property','Value',...) creates a new CC_METHOD_ADV_GUI or
%   raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before CC_method_adv_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to CC_method_adv_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help CC_method_adv_GUI

% Last Modified by GUIDE v2.5 31-May-2005 14:15:08

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @CC_method_adv_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @CC_method_adv_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before CC_method_adv_GUI is made visible.
function CC_method_adv_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
```

Appendix B

```

% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CC_method_adv_GUI (see VARARGIN)

% Choose default command line output for CC_method_adv_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CC_method_adv_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure2);
set(hObject,'Position',[0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = CC_method_adv_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[file_name,path]=uigetfile('C:\*.txt','Open File');    % From Line 88 to 93; allow User to input data
file
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

% From Line 96 to 97; initialised file name of data file
name=strcat(path,file_name);
set(handles.text3 , 'String',name);

% From Line 100 to 123; set visibility of the display of the program for the user to see
set(handles.text14,'visible','on');
set(handles.pushbutton8,'visible','on');
set(handles.pushbutton9,'visible','on');

set(handles.pushbutton2,'visible','off');
set(handles.pushbutton3,'visible','off');
set(handles.edit1,'visible','off');
set(handles.edit2,'visible','off');
set(handles.edit3,'visible','off');

```

Appendix B

```

set(handles.edit4,'visible','off');
set(handles.axes1,'visible','off');
set(handles.axes2,'visible','off');
set(handles.text4,'visible','off');
set(handles.text5,'visible','off');
set(handles.text6,'visible','off');
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
set(handles.text13,'visible','off');
set(handles.text16,'visible','off');
set(handles.text17,'visible','off');

% From Line 126 to 129; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
W = get(handles.edit1,'String')

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');

```


Appendix B

```

else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
L = get(handles.edit2 , 'String')

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Extraction of Vth using (constant current)CC method
% CC method define Vth,cc as the Vgs-value, for which normalized Ids reaches a certain
magnitude
% Very often Ids=W/L*10^-7 is chosen as the arbitrary value
%
%Copyright (c) 2005 Aw Wei Kwan
%date: 10/05/05
%
%

c="";    % clear display value

name=get(handles.text3 , 'String');    % initialise name to user selected choice

[Ax,Bx,Cx,Dx,Ex,Fx,Gx]=textread(name, '%d%f%f%f%f%f', 'headerlines',5);    % start to load
from 5th row of data file
EXTRACTED_DATAS=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

set(handles.text12, 'string',c);    % clear linear Vth value
set(handles.text13, 'string',c);    % clear saturated Vth value
W = str2num(get(handles.edit1 , 'String'));    % initialise width to the value entered by the user
L = str2num(get(handles.edit2 , 'String'));    % initialise length to the value entered by the user
startpt = str2num(get(handles.edit3 , 'String'));    % initialise start point entered by the user for
interpolation
endpt= str2num(get(handles.edit4 , 'String'));    % initialise end point entered by the user for
interpolation

vgs=unique(EXTRACTED_DATAS(:,3));    % get vgs
vds=unique(EXTRACTED_DATAS(:,2));    % get vds

% From Line 214 to 218; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);

```

Appendix B

```

cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATAS(find(EXTRACTED_DATAS(:,3)==vgs(k) &
EXTRACTED_DATAS(:,2)==vds(i)),4);
    end
    if(i==1) % for 0.05 v, the following commands will be executed
        axes(handles.axes1);
        hold on;
        plot(vgs,log(Ids(i,:)),'-r'); % plot graph for lin Id
        title('{CC extraction for lin Ids}','FontSize',10)
        xlabel('Vgs'); % label x-axis
        ylabel('Log Ids'); % label y-axis
        zoom on; % allow user to zoom in and out of the graph
    elseif(i==2) % for 1.0 v, the following commands will be executed
        axes(handles.axes2);
        hold on;
        plot(vgs,log(Ids(i,:)),'-b'); % plot graph for sat Id
        title('{CC extraction for sat Ids}','FontSize',10)
        xlabel('Vgs'); % label x-axis
        ylabel('Log Ids'); % label y-axis
        zoom on; % allow user to zoom in and out of the graph
    end
end

for i=1:length(vds)
    vgsx=startpt:0.00001:endpt; % set start / accuracy / end
    Idsx=interp2(vgs,vds,Ids,vgsx,vds(i),'spline'); % interpolation of the graph

    compareval=W/L*1e-7 % set arbitrary value for CC comparison
    Index1=find(Idsx >= compareval); % search for the value where Ids is more than compareval
    Vth(i)=vgsx(Index1(1)); % store the corresponding Vgs value in Vth
end

The_Threshold_voltage_of_Linear_Id = Vth(1); % Linear Threshold Voltage
The_Threshold_voltage_of_Sat_Id = Vth(2); % Saturated Threshold Voltage

set(handles.text12,'String',The_Threshold_voltage_of_Linear_Id); % display Linear
Threshold Voltage to user
set(handles.text13,'String',The_Threshold_voltage_of_Sat_Id); % display Saturated
Threshold Voltage

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Extraction of Vth using (constant current)CC method
% CC method define Vth,cc as the Vgs-value, for which normalized Ids reaches a certain
magnitude
% Very often Ids=W/L*10^7 is chosen as the arbitrary value
%
```

Appendix B

```

%Copyright (c) 2005 Aw Wei Kwan
%date: 10/05/05
%
%

c="";    % clear comments

name=get(handles.text3 , 'String');    % initialise name to user selected choice

[Ax,Bx,Cx,Dx,Ex,Fx,Gx]=textread(name, '%d%f%f%f%f%f%f', 'headerlines',5);    % start to load
from 5th row of data file
EXTRACTED_DATAS=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

set(handles.text12, 'string',c);    % clear linear Vth value
set(handles.text13, 'string',c);    % clear Saturated Vth value
W = str2num(get(handles.edit1 , 'String'));    % initialise width to the value entered by the user
L = str2num(get(handles.edit2 , 'String'));    % initialise length to the value entered by the user
startpt = str2num(get(handles.edit3 , 'String'));    % initialise start point entered by the user for
interpolation
endpt= str2num(get(handles.edit4 , 'String'));    % initialise end point entered by the user for
interpolation

vgs=unique(EXTRACTED_DATAS(:,3));    % get vgs
vds=unique(EXTRACTED_DATAS(:,2));    % get vds

% From Line 295 to 298; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATAS(find(EXTRACTED_DATAS(:,3)==vgs(k) &
EXTRACTED_DATAS(:,2)==vds(i)),4);
    end
    if(i==2)    % for 1.0 v, the following commands will be executed
        axes(handles.axes2);
        hold on;
        plot(vgs,log(Id(i,:)), '-r');    % plot graph for lin Id
        title('CC extraction for SAT Ids','FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    elseif(i==1)    % for 0.05 v, the following commands will be executed
        axes(handles.axes1);
        hold on;
        plot(vgs,log(Id(i,:)), '-b');    % plot graph for sat Id
        title('CC extraction for LIN Ids','FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    end
end
end

```

Appendix B

```

for i=1:length(vds)
    vgsx=startpt:0.00001:endpt;    %set start / accuracy / end
    ldsx=interp2(vgs,vds,lds,vgsx,vds(i),'spline');    % interpolation of the graph

    compareval=-W/L*1e-7;          % set arbitrary value for CC comparison
    Index1=find(ldsx >= compareval); % search for the value where lds is more than compareval
    Vth(i)=vgsx(Index1(1));        % store the corresponding Vgs value in Vth

end

The_Threshold_voltage_of_Linear_lds = Vth(2);    % Linear Threshold Voltage
The_Threshold_voltage_of_Sat_lds = Vth(1);      % Saturated Threshold Voltage

set(handles.text12,'String',The_Threshold_voltage_of_Linear_lds);    % display Linear
Threshold Voltage to user
set(handles.text13,'String',The_Threshold_voltage_of_Sat_lds);      % display Saturated
Threshold Voltage

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%       str2double(get(hObject,'String')) returns contents of edit3 as a double
startpt = get(handles.edit3,'String')

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

```

Appendix B

```

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double
endpt = get(handles.edit4, 'String')

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    pause(1);
    open=welcome_select_page;
    delete(handles.figure2);

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% From Line 411 to 434; set visibility of the display of the program for the user to see
set(handles.text14, 'visible','on');
set(handles.pushbutton8, 'visible','on');
set(handles.pushbutton9, 'visible','off');

set(handles.pushbutton2, 'visible','on');
set(handles.pushbutton3, 'visible','off');
set(handles.edit1, 'visible','on');
set(handles.edit2, 'visible','on');
set(handles.edit3, 'visible','on');
set(handles.edit4, 'visible','on');
set(handles.axes1, 'visible','on');
set(handles.axes2, 'visible','on');
set(handles.text4, 'visible','on');
set(handles.text5, 'visible','on');
set(handles.text6, 'visible','on');
set(handles.text7, 'visible','on');
set(handles.text8, 'visible','on');
set(handles.text9, 'visible','on');
set(handles.text10, 'visible','on');
set(handles.text11, 'visible','on');
set(handles.text12, 'visible','on');
set(handles.text13, 'visible','on');
set(handles.text16, 'visible','on');
set(handles.text17, 'visible','off');

c="";    % clear comments

```

Appendix B

```

name=get(handles.text3, 'String');    % initialise name to user selected choice

[Ax,Bx,Cx,Dx,Ex,Fx,Gx]=textread(name, '%d%f%f%f%f%f%f', 'headerlines',5);    % start to load
from 5th row of data file
EXTRACTED_DATAS=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

vgs=unique(EXTRACTED_DATAS(:,3));    % get vgs
vds=unique(EXTRACTED_DATAS(:,2));    % get vds

%From Line 446 to 449; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATAS(find(EXTRACTED_DATAS(:,3)==vgs(k) &
EXTRACTED_DATAS(:,2)==vds(i)),4);
    end
    if(i==1)    % for 0.05 v, the following commands will be executed
        axes(handles.axes1);
        hold on;
        plothandle=plot(vgs,log(Ids(i,:)), '-r');    % plot graph for lin Id
        title('{CC extraction for lin Ids}', 'FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    elseif(i==2)    % for 1.0 v, the following commands will be executed
        axes(handles.axes2);
        hold on;
        plothandle=plot(vgs,log(Ids(i,:)), '-b');    % plot graph for sat Id
        title('{CC extraction for sat Ids}', 'FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    end
end

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% From Line 484 to 507; set visibility of the display of the program for the user to see
set(handles.text14, 'visible', 'on');
set(handles.pushbutton8, 'visible', 'off');
set(handles.pushbutton9, 'visible', 'on');

set(handles.pushbutton2, 'visible', 'off');

```

Appendix B

```

set(handles.pushbutton3,'visible','on');
set(handles.edit1,'visible','on');
set(handles.edit2,'visible','on');
set(handles.edit3,'visible','on');
set(handles.edit4,'visible','on');
set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.text4,'visible','on');
set(handles.text5,'visible','on');
set(handles.text6,'visible','on');
set(handles.text7,'visible','on');
set(handles.text8,'visible','on');
set(handles.text9,'visible','on');
set(handles.text10,'visible','on');
set(handles.text11,'visible','on');
set(handles.text12,'visible','on');
set(handles.text13,'visible','on');
set(handles.text16,'visible','off');
set(handles.text17,'visible','on');

c="";    % clear comments

name=get(handles.text3,'String');    % initialise name to user selected choice

[Ax,Bx,Cx,Dx,Ex,Fx,Gx]=textread(name,'%d%f%f%f%f%f','headerlines',5);    % start to load
from 5th row of data file
EXTRACTED_DATAS=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

vgs=unique(EXTRACTED_DATAS(:,3));    % get vgs
vds=unique(EXTRACTED_DATAS(:,2));    % get vds

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATAS(find(EXTRACTED_DATAS(:,3)==vgs(k) &
EXTRACTED_DATAS(:,2)==vds(i)),4);
    end
    if(i==1)    % for 0.05 v, the following commands will be executed
        axes(handles.axes1);
        cla;
        hold on;
        plot(vgs,log(Id(i,:)),'-r');    % plot graph for lin Id
        title('{CC extraction for lin Ids}','FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    elseif(i==2) % for 1.0 v, the following commands will be executed
        axes(handles.axes2);
        cla;
        hold on;
        plot(vgs,log(Id(i,:)),'-b');    % plot graph for sat Id
        title('{CC extraction for sat Ids}','FontSize',10)
        xlabel('Vgs');    % label x-axis
        ylabel('Log Ids');    % label y-axis
        zoom on;    % allow user to zoom in and out of the graph
    end
end

```

Appendix B

```
end
```

```
% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(gcf);    % close window
```

```
% -----
```

TD method:

```
function varargout = TD_method_GUI(varargin)
% TD_METHOD_GUI M-file for TD_method_GUI.fig
%   TD_METHOD_GUI, by itself, creates a new TD_METHOD_GUI or raises the existing
%   singleton*.
%
%   H = TD_METHOD_GUI returns the handle to a new TD_METHOD_GUI or the handle to
%   the existing singleton*.
%
%   TD_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in TD_METHOD_GUI.M with the given input arguments.
%
%   TD_METHOD_GUI('Property','Value',...) creates a new TD_METHOD_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before TD_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to TD_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help TD_method_GUI
```

```
% Last Modified by GUIDE v2.5 31-May-2005 13:04:21
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @TD_method_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @TD_method_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```


Appendix B

```

end
% End initialization code - DO NOT EDIT

% --- Executes just before TD_method_GUI is made visible.
function TD_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to TD_method_GUI (see VARARGIN)

% Choose default command line output for TD_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TD_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure9);
set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = TD_method_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Z1=what('MOS_GUI');
% Z2='\data\*.txt';
% Z3=[Z1.path,Z2];
% [file_name,path]=uigetfile(Z3,'Open File');

[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

name=strcat(path,file_name);
set(handles.text3, 'String', name);

```

Appendix B

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
name=get(handles.text3,'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end

count=0;

for i=1:length(vgs)
    if vgs(i)<=0
        count=count+1;
    end
end

Idslow=Ids(1,:);
Idshigh=Ids(2,:);

vgs1=vgs(count:length(vgs));
Idslowpositive=Idslow(count:length(Idslow));
Idshighpositive=Idshigh(count:length(Idshigh));
%-----
step=vgs1(2)-vgs1(1);
%low vth
vgsstep=0:0.0115:1;
gm1low=gradient(Idslowpositive,step);
gm1lowint=interp1(vgs1,gm1low,vgsstep,'spline');
%figure;
%hold on;
%plot(vgs1,gm1low,'r');
%plot(vgsstep,gm1lowint,'k');

%vgsstep1=0:0.03:1
gm1low=gradient(gm1lowint,step);
%gm1lowint=interp1(vgsstep,gm1low,vgsstep1,'spline');
%figure;
%hold on;
%plot(vgsstep,gm1low,'r');
%plot(vgsstep1,gm1lowint,'k');

vgsstep2=0.2:0.001:0.4
gm2low=gradient(gm1low,step);
gm2lowint=interp1(vgsstep,gm2low,vgsstep2,'spline');
%figure;

```

Appendix B

```

%hold on;
axes(handles.axes1);
cla;
plot(vgsstep,gm2low,'r');
xlabel('Vgs');
ylabel('Gm ^ ^');
%plot(vgsstep2,gm2lowint,'k');
zoom on;
%-----
%high vth

vgsstep=0:0.0115:1;
gmhigh=gradient(ldshighpositive,step);
gmhighint=interp1(vgs1,gmhigh,vgsstep,'spline');
%figure;
%hold on;
%plot(vgs1,gmhigh,'r');
%plot(vgsstep,gmhighint,'k');

%vgsstep1=0:0.03:1
gm1high=gradient(gmhighint,step);
%gm1lowint=interp1(vgsstep,gm1low,vgsstep1,'spline');
%figure;
%hold on;
%plot(vgsstep,gm1high,'r');
%plot(vgsstep1,gm1lowint,'k');

vgsstep2=0.2:0.001:0.35
gm2high=gradient(gm1high,step);
gm2highint=interp1(vgsstep,gm2high,vgsstep2,'spline');
%figure;
%hold on;
axes(handles.axes2);
cla;
plot(vgsstep,gm2high,'r');
%plot(vgsstep2,gm2highint,'k');
xlabel('Vgs');
ylabel('Gm ^ ^');

zoom on;
%-----

[lthlow indexlow]=max(gm2lowint);
vthlow=vgsstep2(indexlow);
[lthhigh indexhigh]=max(gm2highint);
vthhigh=vgsstep2(indexhigh);

%vthlow
%vthhigh

%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%msg2=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg2); waitfor(h);
set(handles.text5,'String',vthlow);
set(handles.text7,'String',vthhigh);
% --- Executes on button press in pushbutton3.

```

Appendix B

```

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    open=welcome_select_page;
    delete(handles.figure9)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text3, 'String');
[a b c d e]=textread(name, '%d %f %f %f %f', 'headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end

count=0;

for i=1:length(vgs)
    if vgs(i)<=0
        count=count+1;
    end
end

Idslow=Ids(1,:);
Idshigh=Ids(2,:);

vgs1=vgs(1:count);
Idslowpositive=Idslow(1:count);
Idshighpositive=Idshigh(1:count);
%-----
step=vgs1(2)-vgs1(1);
%low vth
vgsstep=-1:0.0115:0;
gmlow=gradient(Idslowpositive,step);
gmlowint=interp1(vgs1,gmlow,vgsstep, 'spline');
%figure;
%hold on;
%plot(vgs1,gmlow,'r');
%plot(vgsstep,gmlowint,'k');

%vgsstep1=0:0.03:1
gm1low=gradient(gmlowint,step);
%gm1lowint=interp1(vgsstep,gm1low,vgsstep1, 'spline');

```

Appendix B

```

%figure;
%hold on;
%plot(vgsstep,gm1low,'r');
%plot(vgsstep1,gm1lowint,'k');

vgsstep2=-0.4:0.001:-0.1;
gm2low=gradient(gm1low,step);
gm2lowint=interp1(vgsstep,gm2low,vgsstep2,'spline');
%figure;
%hold on;
axes(handles.axes1);
cla;
plot(vgsstep,gm2low,'r');
%plot(vgsstep2,gm2lowint,'k');
xlabel('\Vgs');
ylabel('Gm ^ ^');
zoom on;
%-----
%high vth

vgsstep=-1:0.0115:0;
gmhigh=gradient(ldshighpositive,step);
gmhighint=interp1(vgs1,gmhigh,vgsstep,'spline');
%figure;
%hold on;
%plot(vgs1,gmhigh,'r');
%plot(vgsstep,gmhighint,'k');

%vgsstep1=0:0.03:1
gm1high=gradient(gmhighint,step);
%gm1lowint=interp1(vgsstep,gm1low,vgsstep1,'spline');
%figure;
%hold on;
%plot(vgsstep,gm1high,'r');
%plot(vgsstep1,gm1lowint,'k');

vgsstep2=-0.4:0.001:-0.1;
gm2high=gradient(gm1high,step);
gm2highint=interp1(vgsstep,gm2high,vgsstep2,'spline');
%figure;
%hold on;
axes(handles.axes2);
cla;
plot(vgsstep,gm2high,'r');
%plot(vgsstep2,gm2highint,'k');
xlabel('\Vgs');
ylabel('Gm ^ ^');
zoom on;
zoom on;
%-----

[lthlow indexlow]=max(gm2lowint);
vthlow=vgsstep2(indexlow);
[lthhigh indexhigh]=max(gm2highint);
vthhigh=vgsstep2(indexhigh);

```

Appendix B

```
%vthlow
%vthhigh
```

```
%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%msg2=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg2); waitfor(h);
set(handles.text5,'String',vthlow);
set(handles.text7,'String',vthhigh);
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
delete(gcf);
```

```
% -----
```

GMLE method:

```
function varargout = GMLE_method_GUI(varargin)
% GMLE_METHOD_GUI M-file for GMLE_method_GUI.fig
%   GMLE_METHOD_GUI, by itself, creates a new GMLE_METHOD_GUI or raises the existing
%   singleton*.
%
%   H = GMLE_METHOD_GUI returns the handle to a new GMLE_METHOD_GUI or the handle
%   to
%   the existing singleton*.
%
%   GMLE_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GMLE_METHOD_GUI.M with the given input arguments.
%
%   GMLE_METHOD_GUI('Property','Value',...) creates a new GMLE_METHOD_GUI or raises
%   the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GMLE_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GMLE_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help GMLE_method_GUI
```

```
% Last Modified by GUIDE v2.5 01-Jun-2005 14:32:04
```

```
% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @GMLE_method_GUI_OpeningFcn, ...
    'gui_OutputFcn',  @GMLE_method_GUI_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
```

Appendix B

```

        'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GMLE_method_GUI is made visible.
function GMLE_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GMLE_method_GUI (see VARARGIN)

% Choose default command line output for GMLE_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GMLE_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure4);

set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = GMLE_method_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

```

Appendix B

```

name=strcat(path,file_name);
set(handles.text3,'String',name);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
name=get(handles.text3,'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];
j=1
for i=1:length(dc(:,2))
    if dc(i,2)==1
        dc_1(j,:)=dc(i,:);
        j=j+1;end
end

j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)>=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1;end
end

%Extracting the necessary column of data
ids=(dc_2(:,4));
vgs=dc_2(:,3);

%Calculate the gm
for i=1:length(ids(:,1))-8
gm(i,1)=(224*ids(i+5)-224*ids(i+3)-56*ids(i+6)+56*ids(i+2)+10.67*ids(i+7)-10.67*ids(i+1)-
ids(i+8)+ids(i))/2.8; end

%Calulate gm_tangent
for i=1:length(gm(:,1))-1
gm_tangent(i,1)=(gm(i+1)-gm(i))/0.01; end

%Change array to fit dimensions. length (minus 9) as vgs_1 length should be
%change to fit the dimensions. vgs(plus 5) as first value of gm_tangent
%occurs when vgs is vgs(5).
j=1;
for j=1:length(vgs(:,1))-9;
    vgs_1(j,1)=vgs(j+5,1);
end

%change array to fit dimensions. length(minus 1) as gm_1 length should be
%change to fit the dimensions. gm(plus 1) as first value of gm_tangent
%occurs when gm is gm(1).
j=1;
for j=1:length(gm(:,1))-1;
gm_1(j,1)=gm(j+1,1);

```


Appendix B

```

end

max=0;

%Find max gm_tangent and its corresponding index
for i =1:length(gm_tangent(:,1))
if gm_tangent(i)<0
break ; end
if gm_tangent(i) > max & gm_tangent(i)>0
max=gm_tangent(i); j=i;
end
end

%Assign x/y axis
x1=vgs_1(j);
y1=gm_1(j);
c=y1-max*x1;
%---edited by kai soon-----
x=vgs_1;
y=gm_1;

m= max; c=y1-m*x1;

%Plot y=mx+c
a=m*x+c;
axes(handles.axes1);
cla;
plot(x,a,x,y);
ylabel('Extrapolation');
xlabel('Vgs');
zoom on;
%-----
%Vth obtained
vthlow=-c/max;
%msg1=sprintf('Vth is %g.\n',vth); h=msgbox(msg1); waitfor(h);
set(handles.text5 , 'String',vthlow);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

open=welcome_select_page;
delete(handles.figure4)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(gcf);

```

Appendix B

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text3,'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];
j=1;
for i=1:length(dc(:,2))
    if dc(i,2)==-0.05
        dc_1(j,:)=dc(i,:);
        j=j+1;end
end

j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)<=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1;end
end

%Extracting the necessary column of data
ids=dc_2(:,4);
vgs=dc_2(:,3);

%Calculate the gm
for i=1:length(ids(:,1))-8
    gm(i,1)=(224*ids(i+5)-224*ids(i+3)-56*ids(i+6)+56*ids(i+2)+10.67*ids(i+7)-10.67*ids(i+1)-
    ids(i+8)+ids(i))/2.8; end

%Calculate gm_tangent
for i=1:length(gm(:,1))-1
    gm_tangent(i,1)=(gm(i+1)-gm(i))/0.01; end

%Change array to fit dimensions. length (minus 9) as vgs_1 length should be
%change to fit the dimensions. vgs(plus 5) as first value of gm_tangent
%occurs when vgs is vgs(5).
j=1;
for j=1:length(vgs(:,1))-9;
    vgs_1(j,1)=vgs(j+5,1);
end

%change array to fit dimensions. length(minus 1) as gm_1 length should be
%change to fit the dimensions. gm(plus 1) as first value of gm_tangent
%occurs when gm is gm(1).
j=1;
for j=1:length(gm(:,1))-1;
    gm_1(j,1)=gm(j+1,1);
end

```

Appendix B

```

max=0;

%Find max gm_tangent and its corresponding index
for i =1:length(gm_tangent(:,1))
if gm_tangent(i)>0
break ; end
if gm_tangent(i)< max & gm_tangent(i)<0
max=gm_tangent(i); j=i;
end
end

%Assign x/y axis
x1=vgs_1(j);
y1=gm_1(j);
c=y1-max*x1;
%---edited by kai soon-----
x=vgs_1;
y=gm_1;

m= max;
c=y1-m*x1;

%Plot y=mx+c
a=m*x+c;
axes(handles.axes1);
cla;

plot(x,y);
hold on;
plot(x,a,'r');
ylabel('Extrapolation');
xlabel('Vgs');
zoom on;
%-----
%Vth obtained
vthlow=-c/max;
%msg1=sprintf('Vth is %g.\n',vth); h=msgbox(msg1); waitfor(h);
set(handles.text5,'String',vthlow);
%-----

```

Integration method:

```

function varargout = Integ_method_adv_GUI(varargin)
% INTEG_METHOD_ADV_GUI M-file for Integ_method_adv_GUI.fig
%   INTEG_METHOD_ADV_GUI, by itself, creates a new INTEG_METHOD_ADV_GUI or
%   raises the existing
%   singleton*.
%
%   H = INTEG_METHOD_ADV_GUI returns the handle to a new INTEG_METHOD_ADV_GUI
%   or the handle to
%   the existing singleton*.
%
%   INTEG_METHOD_ADV_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in INTEG_METHOD_ADV_GUI.M with the given input
%   arguments.

```

Appendix B

```

%
%   INTEG_METHOD_ADV_GUI('Property','Value',...) creates a new
INTEG_METHOD_ADV_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Integ_method_adv_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Integ_method_adv_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Integ_method_adv_GUI

% Last Modified by GUIDE v2.5 01-Jun-2005 14:17:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Integ_method_adv_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @Integ_method_adv_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Integ_method_adv_GUI is made visible.
function Integ_method_adv_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Integ_method_adv_GUI (see VARARGIN)

% Choose default command line output for Integ_method_adv_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Integ_method_adv_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure10);

set(hObject, 'Position', [0 0 205.6 56.8]);

```

Appendix B

```

% --- Outputs from this function are returned to the command line.
function varargout = Integ_method_adv_GUI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

name=strcat(path,file_name);
set(handles.text7, 'String',name);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
name=get(handles.text7, 'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];
clear a;
clear b;
clear c;
clear d;
clear e;
%-----get small drain data-----%
i=0;
j=1;

for i=1:length(dc(:,2)) %-----to find the number of Vg values corresponding with Vd= 0.05V
    if dc(i,2)==0.05
        dc_1(j,:)=dc(i,:);
        j=j+1; end
end

i=0;
j=1;

```

Appendix B

```

for i=1:length(dc_1(:,3)) %-----to find the number of Id values corresponding with Vd= 0.05V
    if dc_1(i,3)>=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
    end
vgs_small=dc_2(:,3);
ids_small=dc_2(:,4);

%figure; %-----plot graph

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%integrate Id w.r.t. Vgs for small drain%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
i=0;
for i = 1:length(ids_small)
    %for j =
        %small_drain_integration(i) = trapz(vgs(), ids());
    %end
    if(i==1)
        small_drain_integ_ids(i)= 0;
    else
        small_drain_integ_ids(i)=((ids_small(i)+ids_small(i-1))*(0.01/2)) + small_drain_integ_ids(i-1);
    end
    result_small(i)=vgs_small(i)-2*(small_drain_integ_ids(i))/ids_small(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ids_small=ids_small';
cla;
axes(handles.axes1);

%figure(1);
%hold;
plohandle=plot(ids_small,result_small,'-r');
    xlabel('Ids');
    ylabel('G(V)');
    zoom on;

%=====

[ithlow indexlow]=max(result_small);
    Vthlow=ithlow;

%msg1=sprintf('Vth is %g.\n',Vthlow);
%h=msgbox(msg1);
%waitfor(h);

set(handles.text3 , 'String',Vthlow);
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

205

Appendix B

```

        %small_drain_integration(i) = trapz(vgs(), ids());
    %end
    if(i==1)
        small_drain_integ_ids(i)= 0;
    else
        small_drain_integ_ids(i)=((ids_small(i)+ids_small(i-1))*(0.01/2)) + small_drain_integ_ids(i-1);
    end
    result_small(i)=vgs_small(i)-2*(small_drain_integ_ids(i))/ids_small(i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ids_small=ids_small';
cla;
axes(handles.axes1);

%figure(1);
%hold;
plothandle=plot(-ids_small,-result_small,'-b');
    xlabel('Ids');
    ylabel('G(V)');
    zoom on;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[ithlow indexlow]=max(result_small);
    Vthlow=ithlow;

%msg1=sprintf('Vth is %g.\n',Vthlow);
%h=msgbox(msg1);
%waitfor(h);

set(handles.text3 , 'String',-Vthlow);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Maximum GM method

```

function varargout = MAXGM_method_GUI(varargin)
% MAXGM_METHOD_GUI M-file for MAXGM_method_GUI.fig
%   MAXGM_METHOD_GUI, by itself, creates a new MAXGM_METHOD_GUI or raises the
%   existing
%   singleton*.
%
%   H = MAXGM_METHOD_GUI returns the handle to a new MAXGM_METHOD_GUI or the
%   handle to
%   the existing singleton*.
%
%   MAXGM_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in MAXGM_METHOD_GUI.M with the given input arguments.
%
%   MAXGM_METHOD_GUI('Property','Value',...) creates a new MAXGM_METHOD_GUI or
%   raises the

```


Appendix B

```

%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before MAXGM_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to MAXGM_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MAXGM_method_GUI

% Last Modified by GUIDE v2.5 31-May-2005 13:39:41

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @MAXGM_method_GUI_OpeningFcn, ...
                  'gui_OutputFcn',    @MAXGM_method_GUI_OutputFcn, ...
                  'gui_LayoutFcn',    [], ...
                  'gui_Callback',     []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MAXGM_method_GUI is made visible.
function MAXGM_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to MAXGM_method_GUI (see VARARGIN)

% Choose default command line output for MAXGM_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MAXGM_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure6);

set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = MAXGM_method_GUI_OutputFcn(hObject, eventdata, handles)

```

Appendix B

```
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end
```

```
name=strcat(path,file_name);
set(handles.text2, 'String',name);
```

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
name=get(handles.text2, 'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));
```

```
for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end
```

```
count=0;
```

```
for i=1:length(vgs)
    if vgs(i)<=0
        count=count+1;
    end
end
```

Appendix B

```

ldslow=lds(1,:);
ldshigh=lds(2,:);

vgs1=vgs(count:length(vgs));
ldslowpositive=ldslow(count:length(lds));
ldshighpositive=ldshigh(count:length(lds));
axes(handles.axes5);
cla;
hold on;
plothandle=plot(vgs1,ldslowpositive,'r');
step=vgs1(2)-vgs1(1);
%-----
%vth low
gmlow=gradient(ldslowpositive,step);

[gradientlow indexlow]=max(gmlow);

Clow=ldslowpositive(indexlow)-gradientlow*vgs1(indexlow);
tangentlow=gradientlow*vgs1+Clow;

axes(handles.axes5);
cla;
hold on;
plothandle=plot(vgs1,ldslowpositive,'r');
plothandle=plot(vgs1,tangentlow,'b');
xlabel('Vgs')
ylabel('Gm')
zoom on;
%axes(handles.axes2);
%-----
%vth high
%gmhigh=gradient(ldshighpositive,step);
%gmhigh=gradient(ldshighpositive/step);
%figure;
%plot(vgs1,gmhigh,'k');

%[gradienthigh indexhigh]=max(gmhigh);

%Chigh=ldshighpositive(indexhigh)-gradienthigh*vgs1(indexhigh);
%tangenthigh=gradienthigh*vgs1+Chigh;

%figure;
%hold on;
%plot(vgs1,ldshighpositive,'r');
%plot(vgs1,tangenthigh,'b');

%-----

vthlow0=-Clow/gradientlow;
vthlow=vthlow0-(vds(1)/2);
%vthhigh0=-Chigh/gradienthigh;
%vthhigh=vthhigh0-(vds(2)/2);

%vthlow

```

Appendix B

```

%vthhigh

%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%-msg1=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg1); waitfor(h);

set(handles.text10, 'String',vthlow);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    open=welcome_select_page;
    delete(handles.figure6)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text2, 'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end

count=0;

for i=1:length(vgs)
    if vgs(i)>=0
        count=count+1;
    end
end

Idslow=Ids(1,:);
Idshigh=Ids(2,:);

len=length(vgs)-count;
vgs1=vgs(1:len+1);
Idslowpositive=Idslow(1:len+1);
Idshighpositive=Idshigh(1:len+1);
%axes(handles.axes5);
%cla;
%hold on;

```

Appendix B

```

%plothandle=plot(vgs1,ldslowpositive,'r');

step=-0.01;
%-----
%vth low
gmlow=gradient(ldshighpositive,step);
axes(handles.axes5);
hold on;
plot(vgs1,gmlow,'r');
[gradientlow indexlow]=min(gmlow);

Clow=ldshighpositive(indexlow)+gradientlow*vgs1(indexlow);
tangentlow=-gradientlow*vgs1+Clow;

axes(handles.axes5);
cla;
hold on;
plothandle=plot(vgs1,ldshighpositive,'r');
plothandle=plot(vgs1,tangentlow,'b');
zoom on;
%axes(handles.axes2);
%-----
%vth high
%gmhigh=gradient(ldshighpositive,step);
%gmhigh=gradient(ldshighpositive/step);
%figure;
%plot(vgs1,gmhigh,'k');

%[gradienthigh indexhigh]=max(gmhigh);

%Chigh=ldshighpositive(indexhigh)-gradienthigh*vgs1(indexhigh);
%tangenthigh=gradienthigh*vgs1+Chigh;

%figure;
%hold on;
%plot(vgs1,ldshighpositive,'r');
%plot(vgs1,tangenthigh,'b');

%-----

vthlow0=-Clow/gradientlow;
vthlow=vthlow0+(vds(1)/2);
%vthhigh0=-Chigh/gradienthigh;
%vthhigh=vthhigh0-(vds(2)/2);

%vthlow
%vthhigh

%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%-msg1=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg1); waitfor(h);

set(handles.text10,'String',vthlow);

```

Appendix B

```
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
delete(gcf);
```

```
%-----
```

Match Point method

```
function varargout = MP_method_adv_GUI(varargin)
% MP_METHOD_ADV_GUI M-file for MP_method_adv_GUI.fig
%   MP_METHOD_ADV_GUI, by itself, creates a new MP_METHOD_ADV_GUI or raises the
%   existing
%   singleton*.
%
%   H = MP_METHOD_ADV_GUI returns the handle to a new MP_METHOD_ADV_GUI or the
%   handle to
%   the existing singleton*.
%
%   MP_METHOD_ADV_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in MP_METHOD_ADV_GUI.M with the given input arguments.
%
%   MP_METHOD_ADV_GUI('Property','Value',...) creates a new MP_METHOD_ADV_GUI or
%   raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before MP_method_adv_GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to MP_method_adv_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help MP_method_adv_GUI
```

```
% Last Modified by GUIDE v2.5 31-May-2005 14:09:20
```

```
% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @MP_method_adv_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @MP_method_adv_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
```

Appendix B

```

[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MP_method_adv_GUI is made visible.
function MP_method_adv_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MP_method_adv_GUI (see VARARGIN)

% Choose default command line output for MP_method_adv_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MP_method_adv_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure3);

set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = MP_method_adv_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% From Line 84 to 86; go back to selection page
    pause(1);
    open=welcome_select_page;
    delete(handles.figure3)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

Appendix B

% handles structure with handles and user data (see GUIDATA)

```
[file_name,path]=uigetfile('C:\*.txt','Open File');
```

% From Line 103 to 107; allow User to input data file

```
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end
```

% From Line 110 to 111; initialised file name of data file

```
name=strcat(path,file_name);
set(handles.text9,'String',name);
```

% From Line 114 to 116; set visibility of the display of the program for the user to see

```
set(handles.pushbutton4,'visible','on');
set(handles.pushbutton7,'visible','on');
set(handles.text20,'visible','on');
```

% From Line 119 to 122; to clear the graph

```
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;
```

% From Line 124 to 141; set visibility of the display of the program for the user to see

```
set(handles.text19,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.edit1,'visible','off');
set(handles.edit2,'visible','off');
set(handles.edit3,'visible','off');
set(handles.edit4,'visible','off');
set(handles.axes1,'visible','off');
set(handles.axes2,'visible','off');
set(handles.pushbutton5,'visible','off');
set(handles.pushbutton6,'visible','off');
set(handles.text2,'visible','off');
set(handles.text3,'visible','off');
set(handles.text4,'visible','off');
set(handles.text5,'visible','off');
set(handles.text22,'visible','off');
set(handles.text23,'visible','off');
set(handles.text24,'visible','off');
```

% --- Executes on button press in pushbutton4.

```
function pushbutton4_Callback(hObject, eventdata, handles)
```

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

Appendix B

```

% Extraction of Vth using (Match-point)MP method
% MP extracts the Vth,mp as the point, where Ids deviates form an expenotial relationship.
%
%
%CopyRight (c) 2005 by AW WEI KWAN + KAI SOON
%date: 150505
%
%
name=get(handles.text9, 'String');    % initialise name to user selected choice

% From Line 162 to 183; set visibility of the display of the program for the user to see
set(handles.text9, 'String', name);
set(handles.text19, 'visible', 'on');
set(handles.text10, 'visible', 'on');
set(handles.text11, 'visible', 'on');
set(handles.edit1, 'visible', 'on');
set(handles.edit2, 'visible', 'on');
set(handles.edit3, 'visible', 'on');
set(handles.edit4, 'visible', 'on');
set(handles.axes1, 'visible', 'on');
set(handles.axes2, 'visible', 'on');
set(handles.pushbutton5, 'visible', 'on');
set(handles.text2, 'visible', 'on');
set(handles.text3, 'visible', 'on');
set(handles.text4, 'visible', 'on');
set(handles.text5, 'visible', 'on');

set(handles.text22, 'visible', 'on');
set(handles.text23, 'visible', 'on');
set(handles.text24, 'visible', 'off');

set(handles.pushbutton6, 'visible', 'off');
set(handles.pushbutton7, 'visible', 'off');

set(handles.text4, 'string', '-');    % show dash for Linear Vth display
set(handles.text5, 'string', '-');    % show dash for Saturated Vth display

[Ax,Bx,Cx,Dx,Ex]=textread(name, '%d%f%f%f%f', 'headerlines', 5);    % start to load from 5th row
of data file
EXTRACTED_DATA=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

% From Line 193 to 198; get datas from 0 to 1
j=1;
for i=1:length(EXTRACTED_DATA(:,3))
    if(EXTRACTED_DATA(i,3)>=0)
        EXTRACTED_DATA_1(j,:)=EXTRACTED_DATA(i,:);
        j=j+1;
    end
end

vgs=unique(EXTRACTED_DATA_1(:,3));    % get vgs
vds=unique(EXTRACTED_DATA_1(:,2));    % get vds

% From Line 204 to 207; to clear the graph

```

Appendix B

```

axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATA_1(find(EXTRACTED_DATA_1(:,3)==vgs(k) &
EXTRACTED_DATA_1(:,2)==vds(i)),4);
    end
    if(i==1) % when Vds is 0.05 v, the following commands will be executed
        axes(handles.axes1);
        hold on;
        plothandle=plot(vgs,log(Id(i,:)),'-r'); % plot graph for lin Id
        title('Plot of Log(Lin-Ids) vs Vgs')
        xlabel('Vgs'); % label x-axis
        ylabel('Log(Ids)'); % label y-axis
        zoom on; % allow user to zoom in and out of the graph
    elseif(i==2) % when Vds is 1.0 v, the following commands will be executed
        axes(handles.axes2);
        hold on;
        plothandle=plot(vgs,log(Id(i,:)),'-b'); % plot graph for sat Id
        title('Plot of Log(sat-Ids) vs Vgs')
        xlabel('Vgs'); % label x-axis
        ylabel('Log(Ids)'); % label y-axis
        zoom on; % allow user to zoom in and out of the graph
    end

end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
name=get(handles.text9,'String'); % initialise name to user selected choice

% From Line 245 to 259; set visibility of the display of the program for the user to see
set(handles.text9,'String',name);
set(handles.text19,'visible','on');
set(handles.text10,'visible','on');
set(handles.text11,'visible','on');
set(handles.edit1,'visible','on');
set(handles.edit2,'visible','on');
set(handles.edit3,'visible','on');
set(handles.edit4,'visible','on');
set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.pushbutton5,'visible','on');
set(handles.text2,'visible','on');
set(handles.text3,'visible','on');

```

Appendix B

```

set(handles.text4, 'visible','on');
set(handles.text5, 'visible','on');

name=get(handles.text9, 'String');    % initialise name to user selected choice
[Ax,Bx,Cx,Dx,Ex]=textread(name, '%d%f%f%f%f', 'headerlines',5);    % start to load from 5th row
of data file
EXTRACTED_DATA=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

% From Line 267 to 273; get data from 0 to 1
j=1;
for i=1:length(EXTRACTED_DATA(:,3))
    if(EXTRACTED_DATA(i,3)>=0)
        EXTRACTED_DATA_1(j,:)=EXTRACTED_DATA(i,:);
        j=j+1;
    end
end

vgs=unique(EXTRACTED_DATA_1(:,3));    % get vgs
vds=unique(EXTRACTED_DATA_1(:,2));    % get vds

% From Line 279 to 282; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        lds(i,k)=EXTRACTED_DATA_1(find(EXTRACTED_DATA_1(:,3)==vgs(k) &
EXTRACTED_DATA_1(:,2)==vds(i)),4);
    end
end

for i=1:length(vds)

    ldsL=log(lds(i,:));

    if (i==1)    % when Vds is 0.05 v, the following commands will be executed
        startptl = str2num(get(handles.edit1, 'String')); % get the start points where the user defines for
the best straight line in the plot
        endptl= str2num(get(handles.edit2, 'String'));    % get the end points where the user defines for
the best straight line in the plot
        vgsx=startptl:0.0001:endptl;    % define the range for Vgs
        ldsLx=interp1(vgs,ldsL,vgsx,'linear');    % Linear interpolation of Log lds
        mintar=endptl;    % set the point where the search for the deviation begins
    elseif (i==2)    % when Vds is 1.0 v, the following commands will be executed
        startpts = str2num(get(handles.edit3, 'String'));    % get the start points where the user defines
for the best straight line in the plot

```

Appendix B

```

endpts= str2num(get(handles.edit4 , 'String'));    % get the end points where the user defines
for the best straight line in the plot
vgsx=startpts:0.0001:endpts;
IdsLx=interp1(vgs,IdsL,vgsx,'linear');    % Saturated interpolation of Log Ids
mintar=endpts;    % set the point where the search for the deviation begins
end

vgsxx=0:0.0001:1;    % to split Vgs into more points
IdsLxx=interp1(vgs,IdsL,vgsxx,'spline');    % performs interpolation on Log Ids and gets more
points
grad= (IdsLx((length(IdsLx))-IdsLx(1))/((vgsx((length(vgsx))-vgsx(1))));    % find gradient of the
straight line defined by the user
C=IdsLx(1)-vgsx(1)*grad;    % find y-intercept at x = 0
Ids2=grad*vgsxx+C;    % equation of straight line; y = mx + c

% From Line 321 to 330; finds the deviation of 5% to get Vth
diff = (Ids2 - IdsLxx);

targ=find(vgsxx>=mintar);

for j=1:length(diff)
    if (j<targ)
        diff(j)=0;
        j=j+1;
    end
end

Index1=find(diff >= log(1.05));

Vth(i)=vgsxx(Index1(1));    % gets Vth where the deviation of Ids and the straight line is 5%

if(i==1)    % when Vds is 0.05 v, the following commands will be executed

axes(handles.axes1);
hold on;
zoom on;    % allow user to zoom in and out of the graph
plot(vgsxx,Ids2,'-g');    % plot straight line
plot(vgs,log(Ids(i,:)),'-r');    % plot linear Ids
plot(Vth(i),Ids2(Index1(1)),'go');    % plot the deviation of 5%

title('MP point extraction lin Ids','FontSize',12)
xlabel('Vgs');    % label x-axis
ylabel('Log(Ids)');    % label y-axis
zoom on;    % allow user to zoom in and out of the graph
elseif(i==2)    % when Vds is 1.0 v, the following commands will be executed

axes(handles.axes2);
hold on;
zoom on;    % allow user to zoom in and out of the graph
plot(vgsxx,Ids2,'-g');    % plot straight line
plot(vgs,log(Ids(i,:)),'-b');    % plot saturated Ids
plot(Vth(i),Ids2(Index1(1)),'go');    % plot the deviation of 5%

```

Appendix B

```

title('MP point extraction sat Ids','FontSize',12)
xlabel('Vgs');      % label x-axis
ylabel('Log(Ids));  % label y-axis
zoom on;           % allow user to zoom in and out of the graph
end
end

Threshold_Voltage_lin=Vth(1);    % Linear Threshold Voltage
Threshold_Voltage_sat=Vth(2);    % Saturated Threshold Voltage

set(handles.text4 , 'String' ,Threshold_Voltage_lin);    % display Linear Threshold Voltage to
user
set(handles.text5 , 'String' ,Threshold_Voltage_sat);    % display Saturated Threshold Voltage

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%       str2double(get(hObject,'String')) returns contents of edit1 as a double
startptl = get(handles.edit1 , 'String');

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

Appendix B

end

```
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

endptl = get(handles.edit2 , 'String');

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
startpts = get(handles.edit3 , 'String');

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else
    set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end
```

Appendix B

```

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double

    endpts = get(handles.edit4 , 'String');

% --- Executes during object creation, after setting all properties.

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text9 , 'String');    % initialise name to user selected choice
[Ax,Bx,Cx,Dx,Ex]=textread(name, '%d%f%f%f%f', 'headerlines',5);    % start to load from 5th row
of data file
EXTRACTED_DATA=[Ax,-Bx,-Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

% From Line 489 to 495; get data from 0 to 1
j=1;
for i=1:length(EXTRACTED_DATA(:,3))
    if(EXTRACTED_DATA(i,3)>=0)
        EXTRACTED_DATA_1(j,:)=EXTRACTED_DATA(i,:);
        j=j+1;
    end
end

vgs=unique(EXTRACTED_DATA_1(:,3));    % get vgs
vds=unique(EXTRACTED_DATA_1(:,2));    % get vds

% From Line 501 to 504; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATA_1(find(EXTRACTED_DATA_1(:,3)==vgs(k) &
EXTRACTED_DATA_1(:,2)==vds(i)),4);
    end
end

```

Appendix B

```

for i=1:length(vds)

    IdsL=log(Idsi,:);

    if (i==1)      % when Vds is 0.05 v, the following commands will be executed
        startptl = str2num(get(handles.edit1 , 'String')); % get the start points where the user defines
        for the best straight line in the plot
        endptl= str2num(get(handles.edit2 , 'String')); % get the end points where the user defines
        for the best straight line in the plot
        vgsx=-endptl:0.0001:-startptl; % define the range for Vgs
        IdsLx=interp1(vgs,IdsL,vgsx,'linear'); % Linear interpolation of Log Ids
        mintar=-startptl; % set the point where the search for the deviation begins
    elseif (i==2)
        startpts = str2num(get(handles.edit3 , 'String')); % gets the start points where the user defines
        for the best straight line in the plot
        endpts= str2num(get(handles.edit4 , 'String')); % gets the end points where the user defines
        for the best straight line in the plot
        vgsx=-endpts:0.0001:-startpts;
        IdsLx=interp1(vgs,IdsL,vgsx,'linear'); % Saturated interpolation of Log Ids
        mintar=-startpts; % set the point where the search for the deviation begins
    end

    vgsxx=0:0.0001:1; % to split Vgs into more points
    IdsLxx=interp1(vgs,IdsL,vgsxx,'spline'); % performs interpolation on Log Ids and gets more
    points
    grad= (IdsLx((length(Idsi))-1)-IdsLx(1))/((vgsx((length(vgsx))-1))-vgsx(1)); % find gradient of the
    straight line defined by the user
    C=IdsLx(1)-vgsx(1)*grad; % find y-intercept at x = 0
    Ids2=grad*vgsxx+C; % equation of straight line; y = mx + c

    % From Line 541 to 552; % finds the deviation of 5% to get Vth
    diff = (Ids2 - IdsLxx);

    targ=find(vgsxx>=mintar);

    for j=1:length(diff)
        if (j<targ)
            diff(j)=0;
            j=j+1;
        end
    end

    Index1=find(diff >= log(1.05));

    Vth(i)=vgsxx(Index1(1)); % gets Vth where the deviation of Ids and the straight line is 5%

    if(i==1) % when Vds is 0.05 v, the following commands will be executed

```


Appendix B

```

axes(handles.axes1);
hold on;
zoom on; % allow user to zoom in and out of the graph
plot(-vgsxx,Ids2,'-g'); % plot straight line
plot(-vgs,log(Ids(i,:)),'-r'); % plot linear Ids
plot(-Vth(i),Ids2(Index1(1)),'go'); % plot the deviation of 5%

title('MP point extraction lin Ids','FontSize',12)
xlabel('Vgs'); % label x-axis
ylabel('Log(Ids)'); % label y-axis
zoom on; % allow user to zoom in and out of the graph
elseif(i==2) % when Vds is 1.0 v, the following commands will be executed

axes(handles.axes2);
hold on;
zoom on; % allow user to zoom in and out of the graph
plot(-vgsxx,Ids2,'-g'); % plot straight line
plot(-vgs,log(Ids(i,:)),'-b'); % plot saturated Ids
plot(-Vth(i),Ids2(Index1(1)),'go'); % plot the deviation of 5%

title('MP point extraction sat Ids','FontSize',12)
xlabel('Vgs'); % label x-axis
ylabel('Log(Ids)'); % label y-axis
zoom on; % allow user to zoom in and out of the graph
end
end

Threshold_Voltage_lin=-Vth(1); % Linear Threshold Voltage
Threshold_Voltage_sat=-Vth(2); % Saturated Threshold Voltage

set(handles.text4,'String',Threshold_Voltage_lin); % display Linear Threshold Voltage to
user
set(handles.text5,'String',Threshold_Voltage_sat); % display Saturated Threshold Voltage to
user

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

name=get(handles.text9,'String'); % initialise name to user selected choice

% From Line 609 to 628; set visibility of the display of the program for the user to see
set(handles.text9,'String',name);
set(handles.text19,'visible','on');
set(handles.text10,'visible','on');
set(handles.text11,'visible','on');
set(handles.edit1,'visible','on');

```

Appendix B

```

set(handles.edit2,'visible','on');
set(handles.edit3,'visible','on');
set(handles.edit4,'visible','on');
set(handles.axes1,'visible','on');
set(handles.axes2,'visible','on');
set(handles.pushbutton6,'visible','on');
set(handles.text2,'visible','on');
set(handles.text3,'visible','on');
set(handles.text4,'visible','on');
set(handles.text5,'visible','on');
set(handles.text22,'visible','on');
set(handles.text23,'visible','off');
set(handles.text24,'visible','on');
set(handles.pushbutton4,'visible','off');
set(handles.pushbutton5,'visible','off');

set(handles.text4,'string','-');    % Display dash for the initial Vth
set(handles.text5,'string','-');    % Display dash for the initial Vth

[Ax,Bx,Cx,Dx,Ex]=textread(name,'%d%f%f%f%f','headerlines',5);    % start to load from 5th row
of data file
EXTRACTED_DATA=[Ax,Bx,Cx,Dx,Ex];    % initialise extracted data variables from user
selected data file

% From Line 637 to 643; gets data from the range 0 to 1
j=1;
for i=1:length(EXTRACTED_DATA(:,3))
    if(EXTRACTED_DATA(i,3)<=0)
        EXTRACTED_DATA_1(j,:)=EXTRACTED_DATA(i,:);
        j=j+1;
    end
end

vgs=unique(EXTRACTED_DATA_1(:,3));    % get vgs
vds=unique(EXTRACTED_DATA_1(:,2));    % get vds

% From Line 649 to 652; to clear the graph
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=EXTRACTED_DATA_1(find(EXTRACTED_DATA_1(:,3)==vgs(k) &
EXTRACTED_DATA_1(:,2)==vds(i)),4);
    end
    if(i==2)    % when Vds is 1.0 v, the following commands will be executed
        axes(handles.axes1);
        hold on;
        plot(vgs,log(Id),'-b');    % plot graph for SAT Id
        title('Plot of Log(Lin-Ids) vs Vgs')
        xlabel('Vgs');    % label x-axis
        ylabel('Log(Ids)');    % label y-axis
    end
end

```

Appendix B

```

        zoom on;           % allow user to zoom in and out of the graph
elseif(i==1)             % when Vds is 0.05 v, the following commands will be executed
    axes(handles.axes2);
    hold on;
    plot(vgs,log(ids(i,:)),'-r'); % plot graph for LIN Id
    title('Plot of Log(sat-Ids) vs Vgs')
    xlabel('Vgs');           % label x-axis
    ylabel('Log(ids)');      % label y-axis
    zoom on;                 % allow user to zoom in and out of the graph
end

```

```
end
```

```

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(gcf); % close window

```

```
% =====
```

Normalized Mutual Integral Difference Method

```

function varargout = NMID_method_GUI(varargin)
% NMID_METHOD_GUI M-file for NMID_method_GUI.fig
%   NMID_METHOD_GUI, by itself, creates a new NMID_METHOD_GUI or raises the existing
%   singleton*.
%
%   H = NMID_METHOD_GUI returns the handle to a new NMID_METHOD_GUI or the handle
%   to the existing singleton*.
%
%   NMID_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in NMID_METHOD_GUI.M with the given input arguments.
%
%   NMID_METHOD_GUI('Property','Value',...) creates a new NMID_METHOD_GUI or raises
%   the existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before NMID_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to NMID_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```
% Edit the above text to modify the response to help NMID_method_GUI
```

```
% Last Modified by GUIDE v2.5 27-May-2005 11:19:21
```

Appendix B

```

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @NMID_method_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @NMID_method_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before NMID_method_GUI is made visible.
function NMID_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to NMID_method_GUI (see VARARGIN)

% Choose default command line output for NMID_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes NMID_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure10);
set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = NMID_method_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

Appendix B

```

%
%CopyRight (c) 2005 Tan Kai Soon
%date: 15/05/05

[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

name=strcat(path,file_name);
set(handles.text7, 'String',name);

set(handles.pushbutton4, 'visible','off');
set(handles.pushbutton2, 'visible','off');
set(handles.pushbutton6, 'visible','on');
set(handles.pushbutton5, 'visible','on');
set(handles.text3, 'String','-');
set(handles.text4, 'String','-');
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text7, 'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];

%-----get small drain data
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==0.05;
        dc_1(j,:)=dc(i,:);
        j=j+1; end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)>=0

```

Appendix B

```

        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end

vgs_smallo=dc_2(:,3);
ids_smallo=dc_2(:,4);
%-----get large drain data
clear dc_1;
clear dc_2;
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==1
        dc_1(j,:)=dc(i,:);
        j=j+1;
    end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)>=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
vgs_bigo=dc_2(:,3);
ids_bigo=dc_2(:,4);

clear dc_1;
clear dc_2;
clear i;
clear j;

%-----intepolation-----
vgs_small=0:0.0001:1;
vgs_big=0:0.0001:1;
ids_small=interp1(vgs_smallo,ids_smallo,vgs_small,'spline'); %interpolation of the graph
ids_big=interp1(vgs_bigo,ids_bigo,vgs_big,'spline'); %interpolation of the graph

%-----plot graph
axes(handles.axes1);
cla;
plothandle=plot(vgs_small,ids_small,'-r');
xlabel('Vgs');
ylabel('P (X)');
zoom on;
axes(handles.axes2);
cla;
plothandle=plot(vgs_big,ids_big,'-r');
xlabel('Vgs');
ylabel('P (X)');

```

229

Appendix B

```
[ithlow indexlow]=max(small_P_result);
vthlow=vgs_small(indexlow);
```

```
[ithhigh indexhigh]=max(big_P_result);
vthhigh=vgs_big(indexhigh);
```

```
set(handles.text3 , 'String',vthlow);
set(handles.text4 , 'String',vthhigh);
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
open=welcome_select_page;
delete(handles.figure10)
```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
name=get(handles.text7, 'String');
[a b c d e]=textread(name, '%d %f %f %f %f', 'headerlines',5);
dc=[a b c d e];
```

```
%-----get small drain data
```

```
i=0;
j=1;
```

```
for i=1:length(dc(:,2))
    if dc(i,2)==-0.05;
        dc_1(j,:)=dc(i,:);
        j=j+1; end
end
```

```
i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)<=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
```

Appendix B

```

vgs_smallo=dc_2(:,3);
vgs_smallo = -vgs_smallo;
ids_smallo=dc_2(:,4);
ids_smallo = -ids_smallo;
%-----get large drain data
clear dc_1;
clear dc_2;
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==-1
        dc_1(j,:)=dc(i,:);
        j=j+1;
    end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)<=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
vgs_bigo=dc_2(:,3);
vgs_bigo=-vgs_bigo;
ids_bigo=dc_2(:,4);
ids_bigo=-ids_bigo;

clear dc_1;
clear dc_2;
clear i;
clear j;
%-----intepolation-----
vgs_small=0:0.0001:1;
vgs_big=0:0.0001:1;
ids_small=interp1(vgs_smallo,ids_smallo,vgs_small,'spline'); %interpolation of the graph
ids_big=interp1(vgs_bigo,ids_bigo,vgs_big,'spline'); %interpolation of the graph

%-----plot graph
axes(handles.axes1);
cla;
plothandle=plot(-vgs_small,-ids_small,'-r');
xlabel('Vgs');
ylabel('P (X)');
zoom on;
axes(handles.axes2);
cla;
plothandle=plot(-vgs_big,-ids_big,'-r');
xlabel('Vgs');
ylabel('P (X)');
zoom on;

```

233

Appendix B

```

vthlow=vgs_small(indexlow);
vthlow=-1*vthlow

[ithhigh indexhigh]=max(big_P_result);
vthhigh=vgs_big(indexhigh);
vthhigh=-1*vthhigh

set(handles.text3 , 'String',vthlow);
set(handles.text4 , 'String',vthhigh);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pushbutton6,'visible','off');
name=get(handles.text7, 'String');
[a b c d e]=textread(name, '%d %f %f %f %f', 'headerlines',5);
dc=[a b c d e];

%-----get small drain data
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==0.05;
        dc_1(j,:)=dc(i,:);
        j=j+1; end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)>=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
vgs_small=dc_2(:,3);
ids_small=dc_2(:,4);
%-----get large drain data
clear dc_1;
clear dc_2;
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==1
        dc_1(j,:)=dc(i,:);
        j=j+1;
    end
end

i=0;
j=1;

```

Appendix B

```

for i=1:length(dc_1(:,3))
    if dc_1(i,3)>=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
vgs_big=dc_2(:,3);
ids_big=dc_2(:,4);

clear dc_1;
clear dc_2;
clear i;
clear j;
%-----plot graph
axes(handles.axes1);
cla;
plothandle=plot(vgs_small,ids_small,'-r');
xlabel('Vgs')
ylabel('Ids')
axes(handles.axes2);
cla;
plothandle=plot(vgs_big,ids_big,'-r');
xlabel('Vgs')
ylabel('Ids')
set(handles.pushbutton2,'visible','on');
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text7,'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];

%-----get small drain data
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==-0.05;
        dc_1(j,:)=dc(i,:);
        j=j+1; end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)<=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end

```

Appendix B

```

vgs_small=dc_2(:,3);
vgs_small = -vgs_small;
ids_small=dc_2(:,4);
ids_small = -ids_small;
%-----get large drain data
clear dc_1;
clear dc_2;
i=0;
j=1;

for i=1:length(dc(:,2))
    if dc(i,2)==-1
        dc_1(j,:)=dc(i,:);
        j=j+1;
    end
end

i=0;
j=1;
for i=1:length(dc_1(:,3))
    if dc_1(i,3)<=0
        dc_2(j,:)=dc_1(i,:);
        j=j+1; end
end
vgs_big=dc_2(:,3);
vgs_big=-vgs_big;
ids_big=dc_2(:,4);
ids_big=-ids_big;

clear dc_1;
clear dc_2;
clear i;
clear j;
%-----plot graph
axes(handles.axes1);
cla;
plothandle=plot(-vgs_small,-ids_small,'-r');
xlabel('Vgs')
ylabel('Ids')
axes(handles.axes2);
cla;
plothandle=plot(-vgs_big,-ids_big,'-r');
xlabel('Vgs')
ylabel('Ids')
%-----
set(handles.pushbutton4,'visible','on');
set(handles.pushbutton5,'visible','off');

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)

```

Appendix B

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

delete(gcf);

%=====

SDL method:
function varargout = SDL_method_GUI(varargin)
% SDL_METHOD_GUI M-file for SDL_method_GUI.fig
%   SDL_METHOD_GUI, by itself, creates a new SDL_METHOD_GUI or raises the existing
%   singleton*.
%
%   H = SDL_METHOD_GUI returns the handle to a new SDL_METHOD_GUI or the handle to
%   the existing singleton*.
%
%   SDL_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in SDL_METHOD_GUI.M with the given input arguments.
%
%   SDL_METHOD_GUI('Property','Value',...) creates a new SDL_METHOD_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before SDL_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to SDL_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help SDL_method_GUI

% Last Modified by GUIDE v2.5 01-Jun-2005 09:34:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @SDL_method_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @SDL_method_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before SDL_method_GUI is made visible.

```


Appendix B

```

function SDL_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to SDL_method_GUI (see VARARGIN)

% Choose default command line output for SDL_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes SDL_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure8);

set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = SDL_method_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file_name,path]=uigetfile('C:\*.txt','Open File');
if (file_name==0)
    disp('you have not selected any file for input into software');
    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

name=strcat(path,file_name);
set(handles.text3, 'String',name);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
name=get(handles.text3, 'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));

```

Appendix B

```

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end

count=0;

for i=1:length(vgs)
    if vgs(i)<=0
        count=count+1;
    end
end

Idslow=Ids(1,:);
Idshigh=Ids(2,:);

vgs1=vgs(count:length(vgs));
Idslowpositive=Idslow(count:length(Idslow));
Idshighpositive=Idshigh(count:length(Idshigh));

step=vgs1(2)-vgs1(1);
%-----
% vth low
logIdslow=log(Idslowpositive);
gm1low=gradient(logIdslow,step);
gm1low=gradient(gm1low,step);
vgsstep=0:0.0115:1;
gm1lowint=interp1(vgs1,gm1low,vgsstep,'spline');
axes(handles.axes1);
cla;
%figure;
%hold on;
%plot(vgs1,gm1low,'r');
plot(vgsstep,gm1lowint,'k');

xlabel('Vgs');
ylabel('Gm ^');
zoom on;
%-----
% vth high
logIdshigh=log(Idshighpositive);
gm1high=gradient(logIdshigh,step);
gm1high=gradient(gm1high,step);
vgsstep2=0:0.0115:1;
gm1highint=interp1(vgs1,gm1high,vgsstep2,'spline');
%figure;
%hold on;
axes(handles.axes2);
cla;
%plot(vgs1,gm1high,'r');
plot(vgsstep2,gm1highint,'k');
xlabel('Vgs');
ylabel('Gm ^');

```

Appendix B

```

zoom on;
%-----

[lthlow indexlow]=min(gm1lowint);
vthlow=vgsstep(indexlow);

[lthhigh indexhigh]=min(gm1highint);
vthhigh=vgsstep2(indexhigh);

vthlow
vthhigh

%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%msg2=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg2); waitfor(h);
set(handles.text5, 'String',vthlow);
set(handles.text7, 'String',vthhigh);
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    open=welcome_select_page;
    delete(handles.figure8)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text3, 'String');
[a b c d e]=textread(name, '%d %f %f %f %f', 'headerlines',5);
DC=[a b c d e];
vds=unique(DC(:,2));
vgs=unique(DC(:,3));

for i=1:length(vds)
    for k=1:length(vgs)
        Ids(i,k)=DC(find(DC(:,2)==vds(i)&DC(:,3)==vgs(k)),4);
    end
end

count=0;

for i=1:length(vgs)
    if vgs(i)<=0
        count=count+1;
    end
end

Idslow=Ids(1,:);
Idshigh=Ids(2,:);

```

Appendix B

```

vgs1=vgs(1:count);
ldslowpositive=ldslow(1:count);
ldshighpositive=ldshigh(1:count);

step=vgs1(2)-vgs1(1);
%-----
% vth low
logldslow=log(ldslowpositive);
gm1low=gradient(logldslow,step);
gm1low=gradient(gm1low,step);
vgsstep=-1:0.015:-0.1;
gm1lowint=interp1(vgs1,gm1low,vgsstep,'spline');
axes(handles.axes1);
cla;
%figure;
%hold on;
%plot(vgs1,gm1low,'r');
plot(vgsstep,gm1lowint,'k');
xlabel('Vgs');
ylabel('Gm ^');
zoom on;
%-----
% vth high
logldshigh=log(ldshighpositive);
gm1high=gradient(logldshigh,step);
gm1high=gradient(gm1high,step);
vgsstep2=-1:0.0115:-0.1;
gm1highint=interp1(vgs1,gm1high,vgsstep2,'spline');
%figure;
%hold on;
axes(handles.axes2);
cla;
%plot(vgs1,gm1high,'r');
plot(vgsstep2,gm1highint,'k');
xlabel('Vgs');
ylabel('Gm ^');
zoom on;
%-----

[lthlow indexlow]=min(gm1lowint);
vthlow=vgsstep(indexlow);

[lthhigh indexhigh]=min(gm1highint);
vthhigh=vgsstep2(indexhigh);

vthlow
vthhigh

%msg1=sprintf('Vth for high voltage is %g.\n',vthhigh); h=msgbox(msg1); waitfor(h);
%msg2=sprintf('Vth for low voltage is %g.\n',vthlow); h=msgbox(msg2); waitfor(h);
set(handles.text5,'String',vthlow);
set(handles.text7,'String',vthhigh);

```

Appendix B

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

delete(gcf);

%=====

Transconductance Change method:

function varargout = TC_method_GUI(varargin)
% TC_METHOD_GUI M-file for TC_method_GUI.fig
%   TC_METHOD_GUI, by itself, creates a new TC_METHOD_GUI or raises the existing
%   singleton*.
%
%   H = TC_METHOD_GUI returns the handle to a new TC_METHOD_GUI or the handle to
%   the existing singleton*.
%
%   TC_METHOD_GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in TC_METHOD_GUI.M with the given input arguments.
%
%   TC_METHOD_GUI('Property','Value',...) creates a new TC_METHOD_GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before TC_method_GUI_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to TC_method_GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help TC_method_GUI

% Last Modified by GUIDE v2.5 01-Jun-2005 14:23:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @TC_method_GUI_OpeningFcn, ...

```

Appendix B

```

        'gui_OutputFcn', @TC_method_GUI_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before TC_method_GUI is made visible.
function TC_method_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to TC_method_GUI (see VARARGIN)

% Choose default command line output for TC_method_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TC_method_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure5);
set(hObject, 'Position', [0 0 205.6 56.8]);

% --- Outputs from this function are returned to the command line.
function varargout = TC_method_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file_name,path]=uigetfile('C:\*.txt','Open File');

if (file_name==0)
    disp('you have not selected any file for input into software');

```

Appendix B

```

    msg1=sprintf('you have not selected any file for input into software');
    h=msgbox(msg1);waitfor(h);
    return;
end

name=strcat(path,file_name);
set(handles.text3,'String',name);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text3,'String');
[a b c d e]=textread(name,'%d %f %f %f %f','headerlines',5);
dc=[a b c d e];
for i=1:length(dc(:,2))    %values from column 2, no. 1 to 302
    if dc(i,2)>0 & dc(i,2)<1 %if each value of column 2 must be greater than 0 or lesser than 1
        dc_1(i,:)=dc(i,:); %if the above condition is satisfied then dc= dc_1
        i=i+1;end
end

j=1;
for i=1:length(dc_1(:,3))    %values from column 3, no. 1 to 151
    if dc_1(i,3)>=0          %if column 3 must be greater or equal to 0
        dc_2(j,:)=dc_1(i,:); %if the above condition is satisfied then dc_1 =dc_2
        j=j+1;end
end

ids=(dc_2(:,4));            %column 4 of dc_2
vds=(dc_2(:,2));            %column 2 of dc_2
for i=1:length(ids(:,1))-8    %column 1 of ids, from no. 1 to 93
    gm(i,1)=(224*ids(i+5)-224*ids(i+3)-56*ids(i+6)+56*ids(i+2)+10.67*ids(i+7)-10.67*ids(i+1)-
    ids(i+8)+ids(i))/2.8; %differentiation of ids using the 8 point method
end

for i=1:length(gm(:,1))-1
    gm_tangent(i,1)=(gm(i+1)-gm(i))/0.01; %finding the derivative of gm
end

vgs=dc_2(:,3); %column 3 of dc_2
j=1;
for j=1:length(vgs(:,1))-9;
    vgs_1(j,1)=vgs(j+1,1);
end

j=1;
for j=1:length(gm(:,1))-1;
    gm_1(j,1)=gm(j+1,1);
end

%Find gm_tangent max

```

Appendix B

```

max=0;

for i =1:length(gm_tangent(:,1))
if gm_tangent(i)<0
break ; end
if gm_tangent(i) > max & gm_tangent(i)>0
max=gm_tangent(i);
j=i;

end
end
axes(handles.axes1);
cla;
plot(vgs_1,gm_tangent);
xlabel('Vgs');
ylabel('Gm tangent');
zoom on;
vthlow=vgs_1(j);
%msg3=sprintf('Vth is %g.\n',vth); h=msgbox(msg3); waitfor(h);

j=1;
for i=1:length(dc(:,2))
if dc(i,2)==1
dc_3(j,:)=dc(i,:);
j=j+1;
end
end

j=1;
for i=1:length(dc_1(:,3))
if dc_3(i,3)>=0
dc_4(j,:)=dc_3(i,:);
j=j+1;
end
end

ids=(dc_4(:,4)); %column 4 of dc_2
vds=(dc_4(:,2)); %column 2 of dc_2
for i=1:length(ids(:,1))-8 %column 1 of ids, from no. 1 to 93
gm1(i,1)=(224*ids(i+5)-224*ids(i+3)-56*ids(i+6)+56*ids(i+2)+10.67*ids(i+7)-10.67*ids(i+1)-
ids(i+8)+ids(i))/2.8; %differentiation of ids using the 8 point method
end

for i=1:length(gm1(:,1))-1
gm_tangent1(i,1)=(gm1(i+1)-gm1(i))/0.01; %finding the derivative of gm
end

vgs1=dc_4(:,3); %column 3 of dc_2
j=1;
for j=1:length(vgs1(:,1))-9;
vgs_2(j,1)=vgs1(j+1,1);
end

j=1;
for j=1:length(gm1(:,1))-1;

```


Appendix B

```

gm_2(j,1)=gm1(j+1,1);
end

max1=0;

for i =1:length(gm_tangent1(:,1))
if gm_tangent1(i)<0
break ; end
if gm_tangent1(i) > max & gm_tangent1(i)>0
max1=gm_tangent1(i);
j=i;
end
end
set(handles.text5 , 'String',vthlow);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

open=welcome_select_page;
delete(handles.figure5)

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

delete(gcf);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

name=get(handles.text3, 'String');
[a b c d e]=textread(name, '%d %f %f %f %f', 'headerlines',5);
dc=[a b c d e];
for i=1:length(dc(:,2))    %values from column 2, no. 1 to 302
    if dc(i,2)<0 & dc(i,2)>-1 %if each value of column 2 must be greater than 0 or lesser than 1
        dc_1(i,:)=dc(i,:); %if the above condition is satisfied then dc= dc_1
        i=i+1;end
end

j=1;
for i=1:length(dc_1(:,3))    %values from column 3, no. 1 to 151
    if dc_1(i,3)<=0          %if column 3 must be greater or equal to 0

```

Appendix B

```

        dc_2(j,:)=dc_1(i,:); %if the above condition is satisfied then dc_1 =dc_2
        j=j+1;end
end

ids=(dc_2(:,4));           %column 4 of dc_2
vds=(dc_2(:,2));           %column 2 of dc_2
for i=1:length(ids(:,1))-8 %column 1 of ids, from no. 1 to 93
    gm(i,1)=(224*ids(i+5)-224*ids(i+3)-56*ids(i+6)+56*ids(i+2)+10.67*ids(i+7)-10.67*ids(i+1)-
    ids(i+8)+ids(i))/2.8; %differentiation of ids using the 8 point method
end

for i=1:length(gm(:,1))-1
    gm_tangent(i,1)=(gm(i+1)-gm(i))/0.01; %finding the derivative of gm
end

vgs=dc_2(:,3); %column 3 of dc_2
j=1;
for j=1:length(vgs(:,1))-9;
    vgs_1(j,1)=vgs(j+1,1);
end

%j=1;
%for j=1:length(gm(:,1))-1;
%gm_1(j,1)=gm(j+1,1);
%end

%Find gm_tangent max
max=0;

for i =1:length(gm_tangent(:,1))
    if gm_tangent(i)>0
        break ; end
    if gm_tangent(i) < max & gm_tangent(i)<0
        max=gm_tangent(i);
        j=i;
    end
end
axes(handles.axes1);
cla;
plot(vgs_1,gm_tangent);
xlabel('Vgs');
ylabel('Gm tangent');
zoom on;
vthlow=vgs_1(j);

set(handles.text5 , 'String',vthlow);

%=====

```