國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

Master Thesis

軟性薄膜電晶體類比電路的良率最佳化

Yield Optimization of Flexible TFT Analog Circuits

魏文恩

Wei, Wen-En

指導教授：李建模 博士

Advisor: Dr. Li, Chien-Mo

中華民國 101 年 11 月

November, 2012

致謝

感謝上帝我畢業了！在研究所的過程當中我學到很多，如今要告別(暫別?)長達18.2年的學生時代，心中百感交集，但卻異常平靜。這篇致謝要感謝很多人，有了許許多多人的幫助才有這篇論文的誕生，謝謝，謝謝！這篇論文的組織如下，第二段先感謝我的指導教授，第三段要感謝我的親友，第四段感謝實驗室的同學們，最後會做一個conclusion。

首先要先謝謝的是我的指導教授李建模教授，老師我想對您說沒有您就沒有今天的我。這兩年多來，老師一直很認真的在教導學生，不僅僅是學業研究，而是連學生做事情的態度、處理事情的方法、待人處事的原則，老師都放在心上。老師謙虛的態度，對研究的熱忱，都深深地影響著學生。感謝老師讓我們有個榜樣可以學習。很幸運能在老師的指導下度過碩士的時光，祝老師身體健康、研究順利。

接下來要感謝我的父母，你們對我無限的關心以及無盡的支持就是我堅持的最大動力。感謝你們總是默默的幫助著我，擔心著我，並給予最大的支持與鼓勵。也辛苦了你們20幾年來的栽培與養育，謝謝你們。爸媽我愛你們。還有辛苦的錦瑤，謝謝妳總是儘可能的幫助我，幫我處理許多瑣事，沒有妳幫我打小怪，我就沒有辦法專心的去打王，辛苦妳了！

最後要感謝實驗室的大家，我們的實驗室的氣氛真的很棒，不管是一起做甚麼事都很歡樂。兄友弟恭，同心協力，情義相挺，不管是認真做研究還是認真休閒，大家都做什麼像什麼，真是我的榜樣。感謝川神，你真的是活動百科全書，你的博學多聞以及幫助學弟們的熱情真的是實驗室的一大支柱。感謝柏瑞給了我的研究許多精闢的建議與幫助，川神離開了你就是我們最大的依靠了。感謝官榆的歌聲還有每次的揪團，辛苦你坐我旁邊常常要回答我一大堆問題，也謝謝你總是願意幫助。我們實驗室的博班學長實在是大家的榜樣與最大的幫助，謝謝你們。接著是同學們，李皓，既然妳不在了我就先講妳，妳真的很酷，非常好相處，謝謝妳的腳步聲帶給我們許多的期待。屌宏，都不想說你了，我一定會想念你的下巴以及耳朵，謝謝你一直是我的好戰友。硬哥，謝謝你把匡匡的FB帶進我們實驗室，你真是我們的好朋友，祝你在RTK還是一樣的硬。有你們這幾個同學真是太

讚了。還有一句話，你們實在走得太早了，生活要學著放慢節奏。接著是威猛的碩二們，Ricky，offer 哥，請以你愛將的身分帶領著大家前進吧！湯摳密，跟你一起奮戰了一年多，我都捨不得離開你了，祝你把 XXX X 回來。啟仁，不要再嗆學長了，尤其是我們敬愛的博班學長們。瑋陞，你那走在時代尖端的軟體們真的是太厲害了，謝謝你坐我後面的時候陪我聊天。車洨，抱歉在這種地方叫你這種名字，謝謝你用 DH 帶領著大家前進，快點免役來 Synopsys 吧，buffer 不要留太多，請小心身體。祝你們大家都能準時畢業。碩一的各位請好好地跟你們優秀的學長們學習吧！祝你們在實驗室的生活都滿意。

能夠畢業真是謝天謝地啊！老師說一句不能成段，所以要再多說幾句，希望進公司之後可以學到很多東西，好好的開發 tool。上不辱父母師長諄諄教誨，下不悖同儕好友殷殷期待。今當遠離，臨表涕泣，不知所云。

摘要

軟性薄膜電晶體技術在近年有越來越多的應用。軟性薄膜電晶體技術相較於傳統的矽技術有許多優點，例如較低的成本、較短的製作時間，以及其可撓的特性。目前軟性薄膜電晶體技術有三個較大的問題：(1) 製程中臨界電壓的變異、(2) 電路老化造成臨界電壓的改變、(3) 電路受到撓曲時漂移率的改變。電路設計者在設計電路時無法有效的考慮各種效應，但以上所提及的效應對電路良率有極大的影響。因此此篇論文提出一個軟性薄膜電晶體類比電路良率最佳化的自動化軟體，考慮以上所提的三種主要效應。此良率最佳化軟體所提出的演算法是建立在反應曲面法(response surface methodology)之上，並利用直交表(orthogonal array)判斷模擬的重要變數。本篇論文使用 8 微米 a-Si 與 10 微米 IGZO 薄膜電晶體製程下四種不同的有機發光二極體驅動電路，以及差動放大器，並且考慮兩種不同的材料技術。實驗結果顯示此最佳化軟體分別提升了 a-Si 與 IGZO 技術發光二極體驅動電路 6.8%、12.0%的良率。此最佳化軟體提供使用者電路中各電晶體寬度的組合以達到在老化、撓曲，以及製程變異的情況下最佳的良率。

關鍵字：良率最佳化、類比電路、軟性薄膜電晶體技術、反應曲面法、直交表

Abstract

Flexible thin-film transistor (TFT) technology is widely used in recent years. Flexible TFT technology has many advantages over conventional silicon technology such as low cost, short manufacturing time, and flexibility. Flexible TFT technology has three important problem that may cause yield loss of flexible TFT circuits. They are 1) Process variation in threshold voltage, 2) aging effect results in threshold voltage shift, and 3) bending effect result in mobility change. It is hard for designer to consider all the effects when designing the flexible TFT circuits. Thus, this thesis proposes a yield optimization automation tool for analog flexible TFT circuits, considering the above three effects. Response surface methodology based optimization flow is proposed in this thesis, using orthogonal array to perform the screening experiment to identify important variables. Organic light-emitting diode (OLED) drivers and operational amplifier (OPAMP) using 8μm amorphous-silicon TFT technology and 10 μm Indium-Gallium-Zinc Oxide TFT technology are demonstrated. Experimental results show that this tool can promote the yield of a-Si and IGZO OLED driver by average 6.8% and 12.0%, respectively.

Keywords: *Yield Optimization, Flexible TFT technology, Analog Circuits, Response Suface Methodology, Orthogonal Array*

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

Flexible electronics *thin-film transistor* (TFT) technologies are widely used in display and has attracted more and more attentions [Jain 05]. Flexible TFT can be manufactured on large flexible substrate. Flexible TFT technology has many advantages, such as low manufacturing cost, light weight, short manufacturing time, and most importantly, flexibility [Hack 89][Kumar 04][Venugopal 07]. The manufacturing cost and time of flexible electronics are significantly lower and shorter than those of the conventional silicon bulk technology thanks to inkjet printing and roll-to-roll manufacturing [Zhou 06][Ishida 10]. Therefore, flexible TFT technologies have many promising applications in flexible display, e-paper, e-book, RFID, *etc* [Nathan 04].

Despite all the above mentioned advantages, current flexible TFT technologies still have three challenges: process variation, performance change due to aging and bending. When flexible TFT circuits are fabricated, process variation occurs and results in a statistical spread on the circuit component parameter value. Thus, process variation may reduce the circuit yield and should be considered when designing flexible TFT circuits. Once the gate bias stress applied to TFTs, it is observed that V_{th} degradation may occur without changing the field effect mobility [Suresh 08]. For the positive gate bias stress, a positive shift in V_{th} (ΔV_{th}) is observed [Cho 11]. For the negative gate bias stress, a negative ΔV_{th} is observed. As the stress time grows, V_{th} degradation gets more severe [Aoki 96], therefore shorten the lifetime of TFTs. In previous reports, two main mechanisms are identified to be responsible for the shift in V_{th}. One is carrier trapping at the channel/dielectric interface, and the other is the creation of additional defect states in the deep-gap states at or near the channel/dielectric interface [Powell 83][Libsch 93][Cho 11]. In Fig 1.1, we can observe that the threshold voltage will shift when operated. It may cause a degradation on the circuit performance. Thus, designers must take aging effect into consideration when designing flexible TFT circuits.

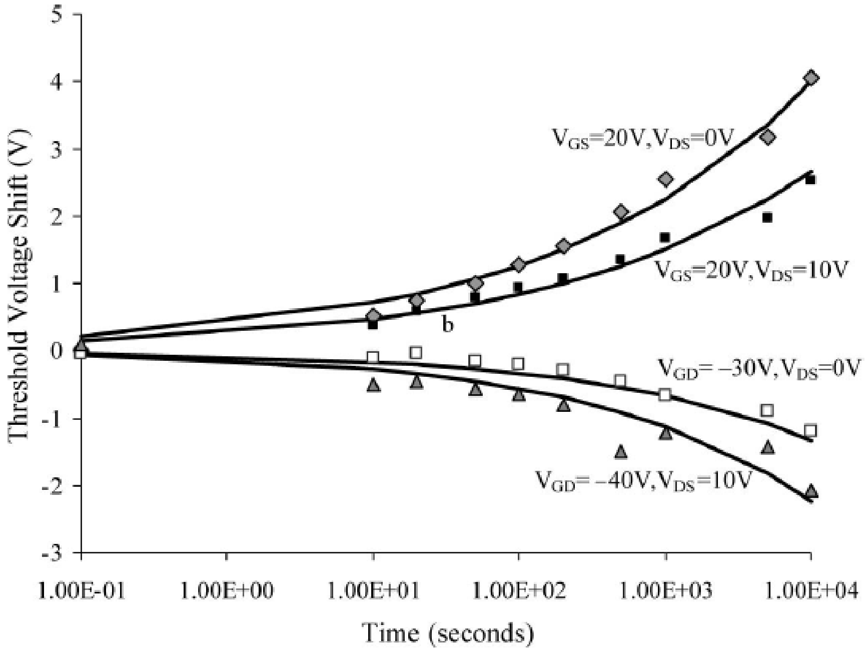


Figure 1.1 V_{th} shift versus stress time at V_{GS} = 20V with different V_{DS} [Shringarpure 07]

According to past research, mobility changes when a mechanical strain is applied to the TFT [Gleskova 02][Sekitani 05a]. Table 1.1 shows the percentage of change in mobility of two different TFT technologies. The range of strains ε (where $\varepsilon = \Delta L/L_0$, with L_0 = initial gate length and ΔL = the change of gate length) are listed in the table. If ε exceeds the range, the TFT will be damaged permanently. We can observe that the response to compressive strain and tensile strain are different in different technologies. The mobility change may be large, therefore affects the circuit performance.

Table 1.1 Percentage of change in mobility of two TFT technologies

TFT technology	Compressive strain	Tensile strain
a-Si TFT [Gleskova 02]	-26% (ε=0%~1%)	7.6% (ε=0%~0.2%)
poly-Si TFT [Cheon 08]	44% (ε=0%~1%)	-44% (ε=0%~1%)
a-IGZO TFT [Munzenrieder 11]	-2% (ε=0%~0.27%)	3% (ε=0%~0.3%)

The above mentioned three important effects – process variation, aging effect, and bending effect affect the yield of flexible TFT circuits significantly. It is difficult for designers to predict so many effects when designing flexible TFT circuits. Besides, circuit optimizer based on traditional bulk silicon technology is not sufficient for flexible TFT circuits because they did not consider bending and aging effects. Therefore, a new EDA tool is needed to automatically adjust TFT sizes to optimize the yield of flexible TFT circuits, considering the process variation, bending effect, and aging effect.

1.2 Proposed Technique

Figure 1.2 shows the flow of the proposed technique, FlexiOptimizer, which is a RSM-based optimization flow. The RSM is the abbreviation of *response surface methodology.* The *YieldAnalyzer* is a modified version of the *yield analysis* proposed in [Ma 11].

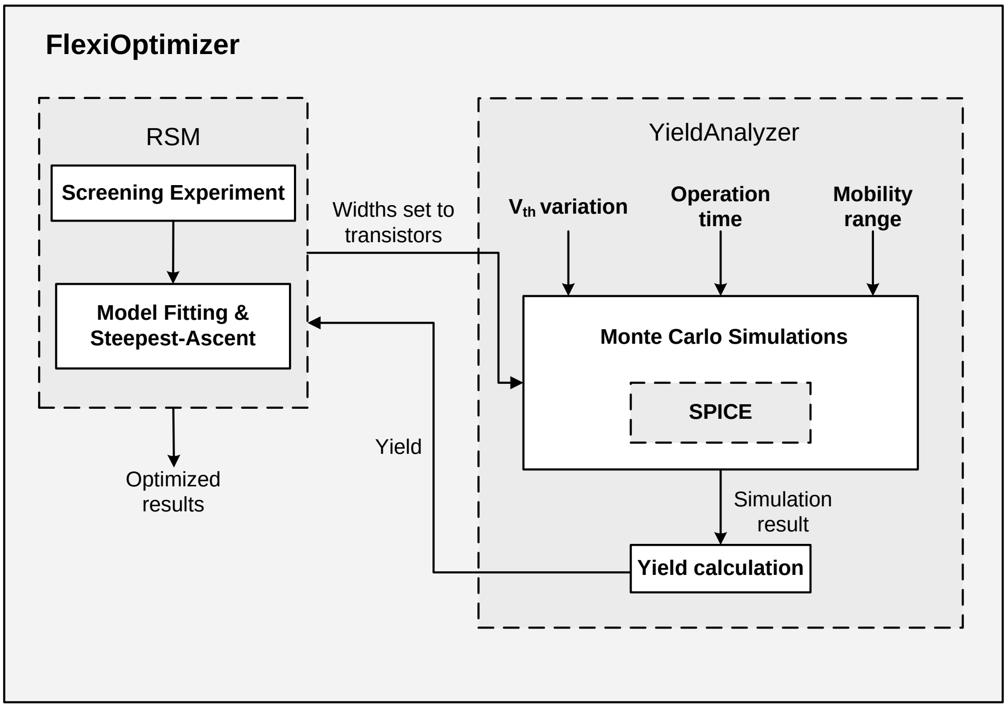


Figure 1.2 Flow of FlexiOptimizer

YieldAnalyzer calculate the yield of flexible TFT circuits considering three important effects: process variation in V_{th}, aging effect, and bending effect. The aging effect can be modeled as the V_{th} change and the bending effect can be modeled as the mobility change. *Monte Carlo simulation* is used to perform the process variation. For process variation in V_{th}, we consider both inter-die variation and intra-die variation. To perform the inter-die variation, an user-specified distribution is taken as the input. To perform the intra-die variation, an user-specified standard deviation (σ) for normal distribution should be given. To consider the bending effect, the highest and the lowest possible value of mobility are used. The circuit should pass both conditions (highest mobility and lowest mobility) or it will be judged as bad circuit. To model the aging effect, the aging model for both amorphous-Silicon (a-Si) and Indium-Gallium-Zinc oxide (IGZO) technologies are used. Given a specified operation time, ΔV_{th} is calculated incrementally. We divide the total time (t_{total}) into several time intervals $t_{interval}$. At the beginning of the i_{th} time interval, $t_{interval,i}$, we perform a DC simulation and calculate the corresponding $\Delta V_{th}(t_{interval,\ i})$. Then we updated the bias of each TFT and continue the next time interval. This process is repeated until we reach the total operation time, t_{total}. The final $\Delta V_{th}(t_{total})$ is then the summation of all $\Delta V_{th}(t_{interval,\ i})$. To calculate the yield, a circuit should pass all the spec(s) both under the *New* (operation time = 0) condition and the *Aged* (after a specified operation time) condition.

The RSM includes two main parts: *screening experiment* and *model fitting &*

steepest-ascent. The screening experiment is used to identify *important variables* at the beginning of the optimization flow. Thus, we can leave fewer variables in the following step of the flow and improve the optimization effectiveness. We use *orthogonal array* (OA) to do the screening experiment. OA is a famous *design of experiment* (DoE) technique, which evenly scatters the values of input variables to describe the response by relatively small number of experiments [Rao46][Rao47][Rao49][Hedayat 99]. After the screening experiment, the model fitting & steepest-ascent begins. This step is to approximate the *response surface* by a first-order model. After a first-model model is fitted, steepest-ascent is used to find the optimal solution [Myers 02][Su 93]. The steepest-ascent starts from the best known solution so far and moves along the steepest direction. The first-order model fitting and steepest-ascent is repeated until the yield cannot be improved anymore. Finally, a second-order model is fitted and the optimal solution is obtained.

Figure 1.3 shows a five-transistor driver used in the experimental results fabricated in 10 μm IGZO technology. Table 1.2 shows the optimized results of the circuit shown in Figure 1.3 with the specifications: 1) 10μm < width of each transistor W_i < 100μm. 2) total width of transistor W_{total} < 300um. 3) OLED current I_{OLED} is within the region 1μA < I_{OLED} < 20μA. 4) width of each transistor is within 10μm and 100μm. The

Monte Carlo trials is 1,000. Operation time is 10,000 seconds. The result is compared with the design according to the view of designers. Since the I_{OLED} should be within 1μA and 20μA. The designer will design the circuit with the I_{OLED} = 10.5μA. The designed W/L is less than 1. Since the ratio is smaller than the lower bound set to the width of each transistor, we set the width to be its minimum feasible value = 10μm. We can observe that the optimized yield is over 20% higher than the designed yield because the designer does not consider the process variation, aging effect, and bending effect.

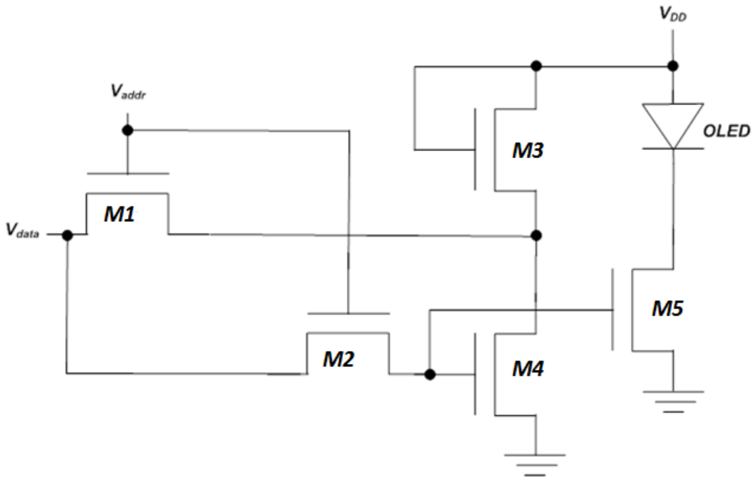


Figure 1.3 Five-transistor OLED driver [Servati 02a]

Table 1.2 Optimization result of circuit in Figure 1.3

	Yield	W1	W2	W3	W4	W5
Designed	81.2%	10μm	10μm	10μm	10μm	10μm
FlexiOptimizer	90.0%	65μm	88μm	42μm	48μm	12μm

1.3 Contributions

This thesis has the following contributions to the research in flexible electronics and yield optimization for flexible TFT circuits.

- Considers process variation, bending effect, and aging effect
- Two technologies – a-Si and IGZO TFT, and two designs – operational amplifier and OLED drivers are demonstrated
- Propose an orthogonal-array-based screening experiment flow that identifies important variables and provides the stability to the optimizer
- DC and AC specs are supported
- Response surface methodology is used to optimize

1.4 Organization

This thesis is organized as follows. Background knowledge is reviewed in chapter 2. Chapter 3 presents the proposed technique – FlexiOptimizer. Chapter 4 shows the experimental results of the proposed technique. Chapter 5 concludes this thesis. Finally, the appendix is the user manual of FlexiOptimizer.

Chapter 2 Background

Chapter 2 reviews background knowledge related to this thesis. Section 2.1 introduces the two popular flexible *thin-film transistor* (TFT) technologies – *amorphous-silicon* (a-Si) TFT and *Indium-Gallium-Zinc oxide* (IGZO) TFT. Section 2.2 summarizes the past researches about the bending effect and the aging effect on TFT. Section 2.3 introduces a method – *orthogonal array* (OA), for planning experiments when the number of input variables is high. Section 2.4 discusses the knowledge of four methods in analog circuit optimization, such as *genetic algorithm* (GA), *geometric programming* (GP), *simulated annealing* (SA), and *response surface methodology* (RSM). Section 2.5 introduces the *FlexiAnalyzer*.

2.1 Flexible TFT Technologies

2.1.1 Amorphous-Silicon Thin-Film Transistors

Amorphous-silicon (a-Si) TFT technology is widely used in large-area and flexible electronics due to its low manufacturing cost, spatial uniformity, and compatibility with large-area process [Hack 89][Kumar 04][Venugopal 07]. In recent years, the most popular application of a-Si TFT technology are pixel circuits for active matrix liquid crystal display (AMLCD) [DaCosta 94] and active matrix organic light-emitting diode (AMOLED) [Servati 02a] [Kumar 05]. As a-Si TFT technology advances, not only pixel circuits but also large-scale analog and digital circuits can be implemented in a-Si TFT technology. Examples of analog circuits include a threshold voltage shift compensating differential amplifier [Madeira 97], an embedded level shifter [Bae 06], and an implementation of an operational amplifier [Tarn 10]. Examples of digital circuits include a pass-transistor logic based multiplexer [Mohan 02], an integrated row driver for high-resolution applications [Kim 06], an asynchronous finite impulse response filter [Bai 08], and pseudo-CMOS design style [Huang 10].

The structure of the amorphous silicon TFT can be roughly classified into staggered and coplanar types according to the location of its patterned active layer. In the staggered type, a gate electrode and source/drain electrode are separated by an interposed semiconductor layer. In the coplanar type, the electrodes are all formed in the same side of a semiconductor layer. The most widely used a-Si TFT is the inverse-staggered type [Jang 99]. Figure 2.1 shows the cross section of an inverse staggered hydrogenated a-Si TFT [Servati 02b, 02c], and it is called bottom-gate structure since the gate is at the bottom of the a-Si TFT. Three films are deposited by plasma enhanced chemical vapor deposition: *hydrogenated amorphous silicon* (a-Si:H), *hydrogenated amorphous silicon nitride* (a-SiN$_x$:H), and *n-doped microcrystalline hydrogenated silicon* (n$^+$μc-Si:H). The thickest layer is the a-SiN$_x$:H layer, which is the gate insulator of the a-Si TFT. When the a-Si TFT is turned on, electrons move across the channel in the a-Si:H layer.

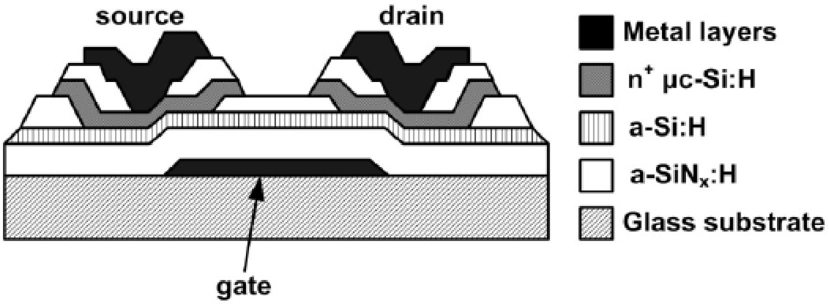


Figure 2.1 Cross section of an a-Si TFT [Servati 02b, 02c]

Figure 2.2 shows the I_D-V_{GS} curve of the a-Si TFT [Servati 02b]. Depending on the bias voltage and device geometry, four regimes of operations are identified: *above threshold*, *forward sub-threshold*, *reverse sub-threshold*, and *Poole-Frenkel emission*. In the Poole-Frenkel emission regime, the front channel conduction is responsible for the leakage current at high negative gate and high positive drain voltages. The conduction path is provided by the accumulation of holes at the top a-Si:H/a-SiN$_x$:H interface, and the holes are generated because of the Poole-Frenkel field-enhanced thermo-ionic emission. In the reverse sub-threshold regime, the leakage current is caused by the back channel conduction due to the weak channel formed at the bottom a-Si:H/a-SiN$_x$:H interface. In the forward sub-threshold regime, the current is caused by interface charges and deep defects in the a-Si:H layer. In the above threshold regime, the a-Si TFT is turned on, and a channel is formed in the a-Si:H layer. The I_D-V_{GS} curve in this regime is similar to that of the MOS transistor.

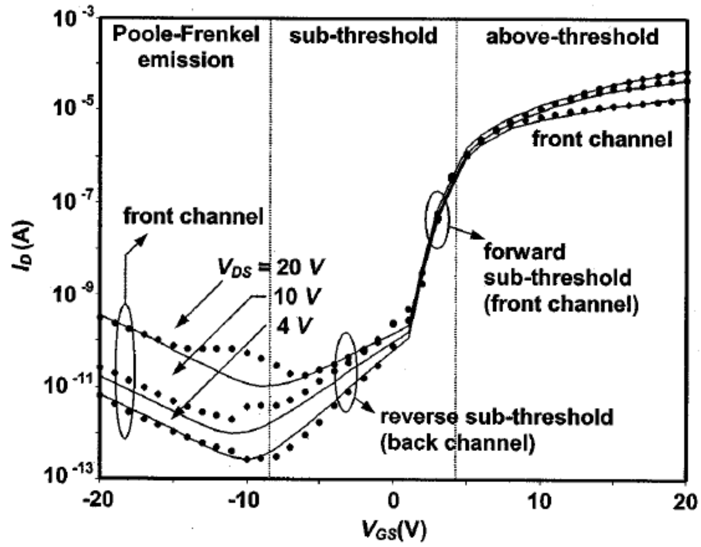


Figure 2.2 I_D-V_{GS} curve of the a-Si TFT [Servati 02b]

2.1.2 Indium-Gallium-Zinc Oxide Thin-Film Transistors

In recent years, *Indium-Gallium-Zinc oxide* (IGZO) thin-film transistor technology is emerging as a promising technology for display and sensor applications. it has attracted much attention due to its atmospheric stability, transparency, low fabrication temperature, and relatively high field-effect mobility [Fortunato 04] [Dehuff 05]

[Nomura 06]. Moreover, inkjet-printing technique can be applied to IGZO TFTs. For example, Kim et al. apply inkjet printing technique to IGZO channel formation. The IGZO ink can be successfully inkjet-printed to create an active channel layer [Kim 09]. IGZO TFT circuits manufactured on flexible substrate can be found in [Park 09][Chien 11][Kim 09].

The common gate, inverted staggered amorphous IGZO (a-IGZO) TFTs is a popular structure for studying the fundamental electrical properties [Fung 10]. Figure 2.3 shows the cross section of an a-IGZO TFT [Fung 10], they selected heavily doped (n^{++}) silicon wafer with 100nm thermal oxide layer as gate electrode and gate dielectric layer, respectively. A 40nm thick a-IGZO active layer was pulse-laser deposited on the substrate. The 50nm thick aluminum (Al) source/drain electrodes were deposited by thermal evaporation. Finally, the device was thermally annealed in air at 300 ˚C for 5 minutes.

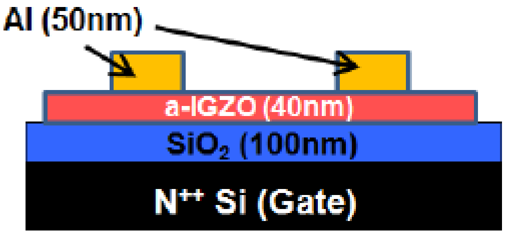


Figure 2.3 cross section of an a-IGZO TFT [Fung 10]

Figure 2.4 shows the output characteristic of the IGZO TFT in the structure of

Figure 2.3. The gate to source voltages (V_{GS}) ranges from 4V to 20V. We can distinct the linear and saturation region clearly. Figure 2.5 shows the linear region (V_{DS}=0.1V) transfer curve of the TFT in the structure of Figure 2.3, the threshold voltage and field effect mobility are extracted based on the standard MOSFET equation (equation(1)) with $V_{DS} << V_{GS}-V_{th}$:

$$I_D = \mu_{eff} C_{ox} \frac{W}{L} [(V_{GS} - V_{th})V_{DS} - \frac{1}{2} {V_{DS}}^2] \cong \mu_{eff} C_{ox} \frac{W}{L} (V_{GS} - V_{th})V_{DS} \tag{1}$$

where C_{ox} is the gate insulator capacitance per unit area, W and L are TFT channel width and length, respectively.

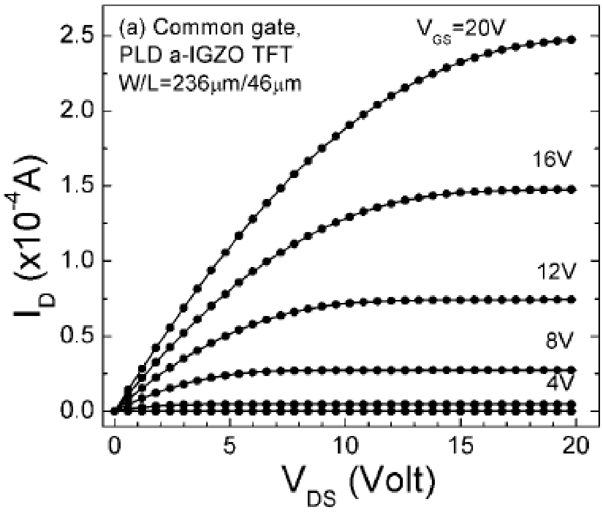


Figure 2.4 I_D-V_{DS} curve of an a-IGZO TFT [Fung 10]

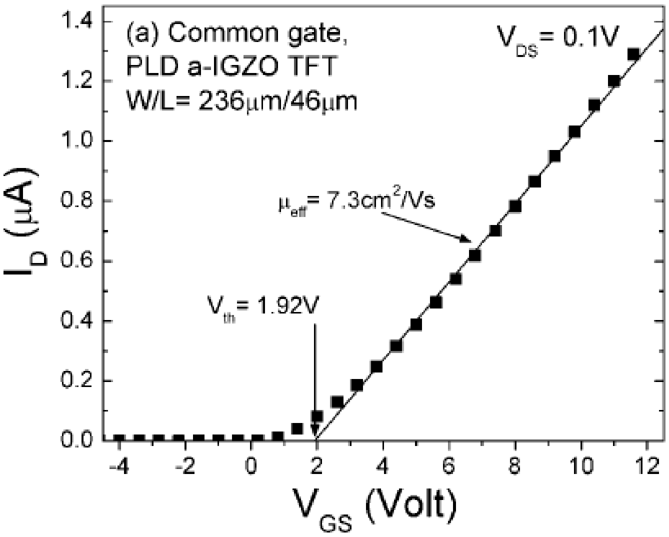


Figure 2.5 I_D-V_{GS} curve of an a-IGZO TFT [Fung 10]

Past research showed the electronic structure and carrier transport mechanism in a-IGZO are similar to that in single crystalline IGZO [Takagi 05].

2.2 Bending Effect and Aging Effect on Flexible TFTs

2.2.1 Bending Effect on Flexible TFTs

According to past research, mobility changes when a mechanical strain is applied to the TFT [Gleskova 02][Sekitani 05a]. Figure 2.6 shows two types of strains: *compressive strain* and *tensile strain*. Compressive strain squeezes the source and drain toward each other as shown in Figure 2.6(a), which is applied to the TFT when the substrate is inward bended. Tensile strain pulls the source and drain away from each other as shown in Figure 2.6(b), which is applied to the TFT when the substrate is outward bended. The dotted circle in Figure 2.6(b) shows the bending radius of curvature *R*.

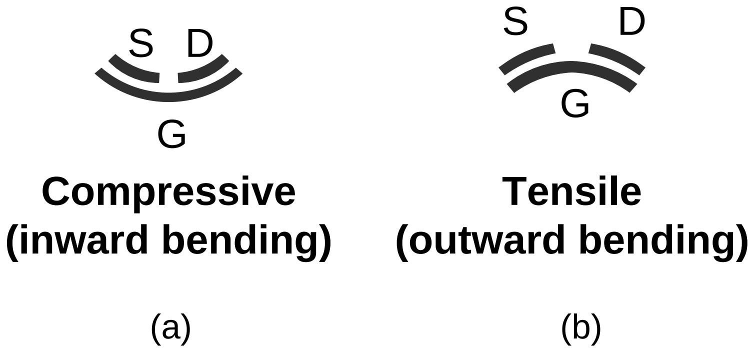


Figure 2.6 (a) compressive strain (b) tensile strain [Sekitani 05a]

Gleskova *et al*. performed a series of experiments to study the electrical performance of a-Si TFTs during and after the application of a strain [Gleskova 99][Gleskova 02]. Figure 2.7 shows the cross section of the a-Si TFT used in their experiments. The thicknesses of the substrate and the film are 25μm and 500nm, and the Young's moduli of the substrate and the film are 5GPa and 183GPa, respectively. The bold arrows in Figure 2.7 represent the bending direction, which is parallel to the current path between the drain and the source.

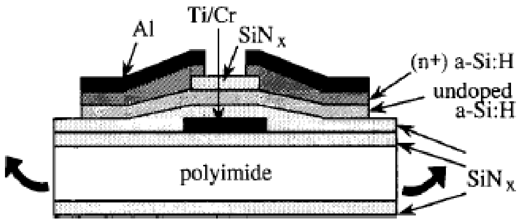


Figure 2.7 Cross section of the a-Si TFT in [Gleskova 02]

Figure 2.8 shows the inward and outward strains as functions of radius of curvature.

The two bold curves represent strains applied to the TFTs with a layer of SiN$_x$ coated at the bottom side of the substrate. The two thin curves represent strains applied to the TFTs with no bottom SiN$_x$ layer, and they are calculated using Equation (2) with d_{f2} equals to 0. It shows that the strain is increased as the radius of curvature decreases.

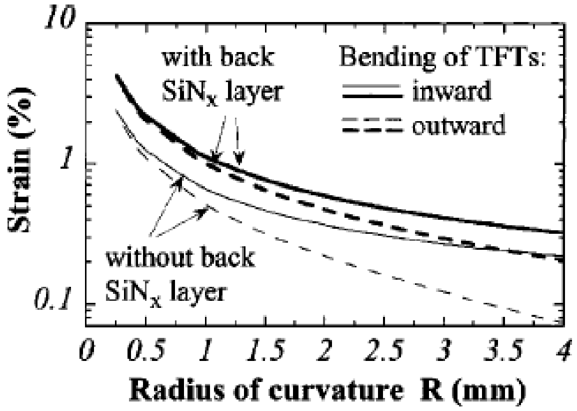


Figure 2.8 Strains as functions of radius of curvature [Gleskova 99]

Two types of experiments are performed in [Gleskova 02]. The first experiment studies the electrical performance of a-Si TFTs after a strain is applied to the TFT, and the second one studies the electrical performance during the application of a strain. In the first experiment, all a-Si TFTs are first bent for one minute, then released and measured. The measured on current I_{on}, source-gate leakage current I_{leak}, threshold voltage V_t, and electron mobility μ are shown in Figure 2.9. The abrupt change in I_{leak} shows TFT failure caused by excessive tensile strain. It is observed that after the strain is applied, TFT characteristic remains the same except the failure ones.

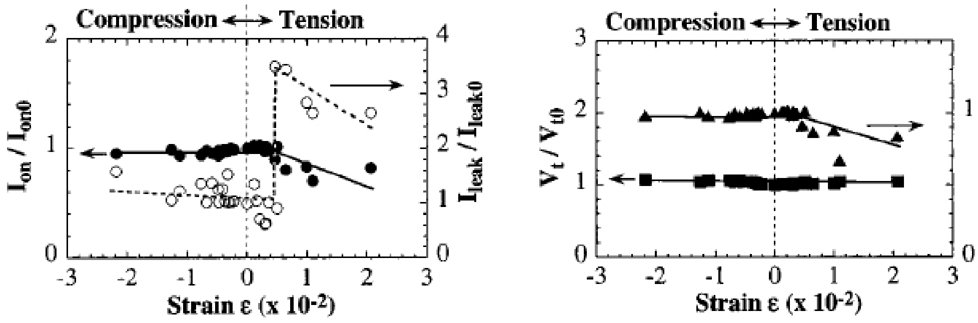


Figure 2.9 Measurement result after the application of a strain [Gleskova 02]

Figure 2.10 shows the relative mobility μ/μ0 of a-Si as a function of ε. μ/μ0 decreases linearly with increasing compressive strain and increases linearly with increasing tensile strain. The relative mobility can be expressed as the linear equation: μ/μ0 = 1+26ε, where ε ranges from -1% to 0.3%. The maximum increase in μ/μ0 is 7.6% and the maximum reduction in μ/μ0 is -26%.

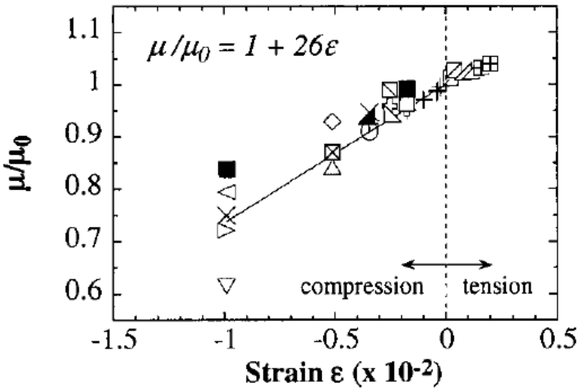


Figure 2.10 Relative mobility as a function of strain [Gleskova 02]

2.2.2 Aging Effect on Flexible TFTs

Once the gate bias stress applied to TFTs, it is observed that threshold voltage (V_{th})

degradation may occur without changing the field effect mobility [Suresh 08]. For the positive gate bias stress, a positive shift in V_{th} (ΔV_{th}) is observed [Cho 11]. For the negative gate bias stress, a negative ΔV_{th} is observed. As the stress time grows, V_{th} degradation gets more severe [Aoki 96], therefore shorten the lifetime of TFTs. The effect is also known as bias stress instability. In previous reports, two main mechanisms are identified to be responsible for the shift in V_{th}. One is carrier trapping at the channel/dielectric interface, and the other is the creation of additional defect states in the deep-gap states at or near the channel/dielectric interface [Powell 83][Libsch 93][Cho 11].

Figure 2.11 (a) shows the two transfer characteristic curves of the same TFT device [Suresh 08]. After the first gate voltages sweep, the gate electrode was stressed at 30V for 500 seconds while V_{DS} was kept at 1V and I_{DS}-V_{GS} sweep was done at V_{DS} = 15V. After this, the second gate voltage sweep was performed. We can see that the transfer characteristic curves [log(I_{DS}) – V_{GS}] of the device before and after the bias stress have similar shapes but a parallel shift along the gate voltage axis between them. The difference between the two curves is the typical threshold voltage shift observed in TFTs after applying bias stress. Figure 2.11 (b) shows the plot of the square root of the drain current as a function of the applied gate bias. We can observe that the slopes of the linear part of the curves, which are proportional to the field effect mobility in the saturation regime, remain unchanged after the 500 seconds bias stress. Thus, we can infer that the extracted mobility remains unaltered after the application of gate bias stress.

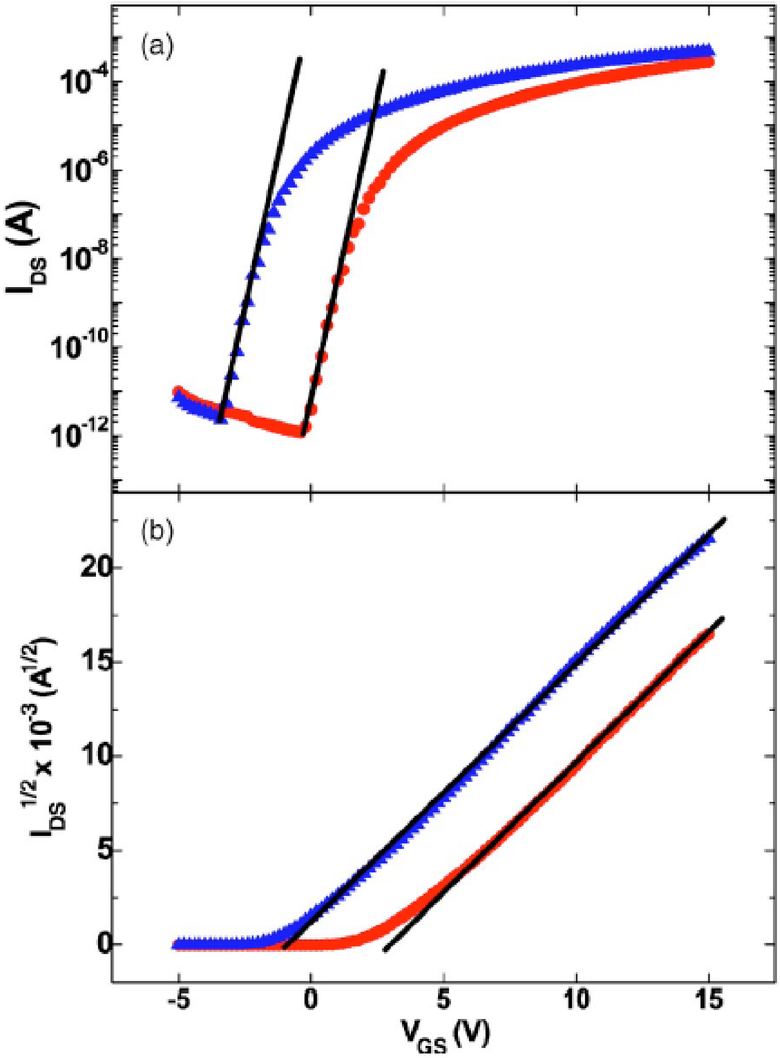


Figure 2.11 Effect of bias stress on the drain current

(▲– prestress, ●– poststress) [Suresh 08]

To better understand the threshold voltage degradation, past research used stretched-exponential model to model the effect [Sung 10][Zan 10][Kim12][Zhang 11]. The standard form of stretched-exponential model [Libsch 93] is

$$\left|\Delta V_{th}\right| = \left|\Delta V_0\right| \{1 - \exp[-(\frac{t_{stress}}{\tau})^{\beta}]\} \qquad (2)$$

where ΔV_0 is the effective voltage drop across the gate insulator, α is the exponent for ΔV_0 dependence, β is the stretched exponential exponent, τ represents the characteristic trapping time of carriers.

In [Shringarpure 07], they proposed an equation based on the stretched-exponential model for a-Si TFT. Figure 2.12 shows the V_{th} shift versus stress time. The model they proposed to fit the curves in Figure 2.12 is as follows:

$$\Delta V_{th}(t) = A \cdot \exp(-\frac{E_A}{kT}) \cdot t^{\beta} (V_{GS} - \eta V_{DS} - V_{TH,0})^{n} \qquad (3)$$

In their simulation results, the model accurately models the V_{th} shift under different bias stress conditions.

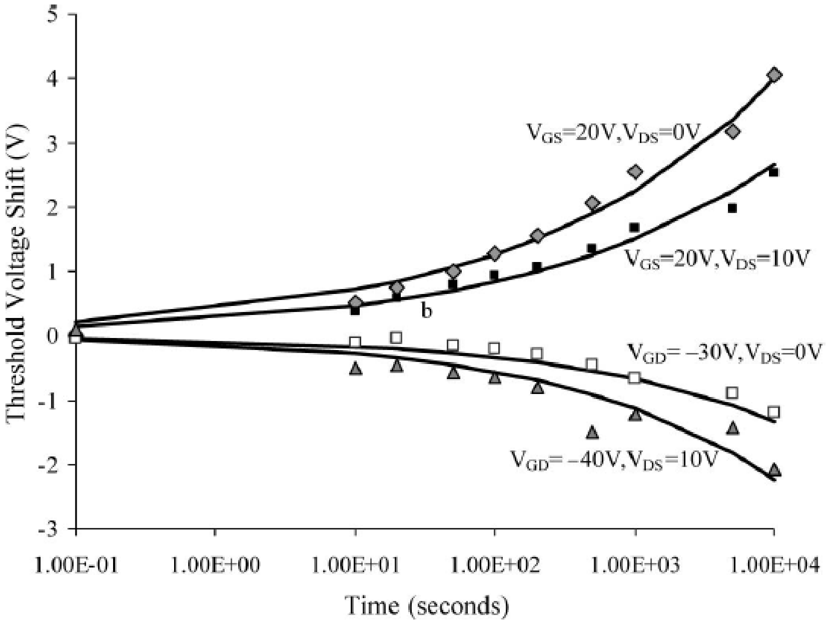


Figure 2.12 V_{th} shift versus stress time at V_{GS} = 20V with different V_{DS} [Shringarpure 07]

2.3 Orthogonal Array

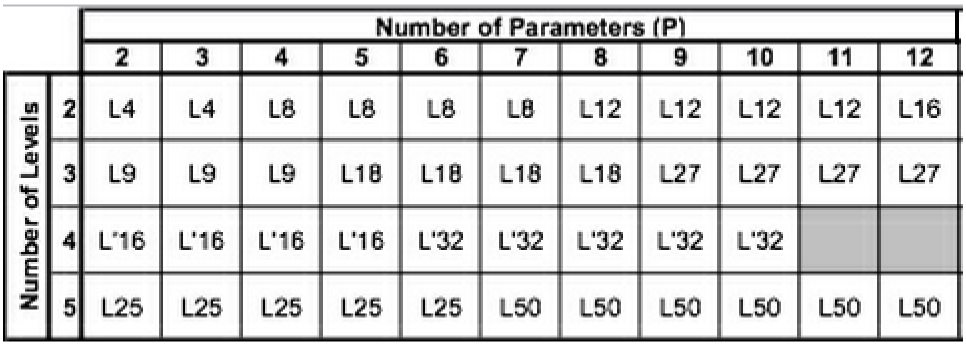
In 1940's, in a series of papers [Rao 46][Rao 47][Rao 49], C. R. Rao introduced the concept of *orthogonal array* (OA). Since the introduction, many researchers in different regions came to this subject and made significant contributions to this field [Hedayat 99]. OA is a very useful method in various regions. It is primarily used in statistics but also can be applied to computer science, cryptography, medicine, agriculture, manufacturing, etc.

OA is often employed in industrial experiments to study the effect of several control variables and is considered to be useful when the number of control variables is high and there are numerous local optima [Tang 93][Leung 01]. When a large number of factors are to be studied in an experiment but only few of them are effective, OA is useful to distinguish effective factors. A special significance of OA is the use in industrial experimentation by [Taguchi 86] to determine the optimum mix of factors to maximize yield and the effect of noise factors such as environmental conditions on production. Since this, OA is often used to optimize the yield of manufacturing when there are many control variables.

Before solving an optimization problem, we usually do not have information about the location of the global optimum. In this case, OA can effectively solve this problem by locating initial design points scattered evenly in the feasible design space so that we can obtain sufficient information in relatively low number of simulations when we start an optimization problem. To define an OA, we must identity 1) number of variables to be studied, 2) levels for each variable. An experimenter should define the number of levels for each variable and the corresponding value for each variable. For example, temperature is now a control variable of an experiment and varies from 288K (absolute degree) to 368K. Experimenter can assign the number of levels for temperature to be 3 (low, medium, and high) and assign the corresponding values to be 308K, 328K, 348K, respectively. If the range of a variable is small, experimenter can use less number of levels, and vice versa.

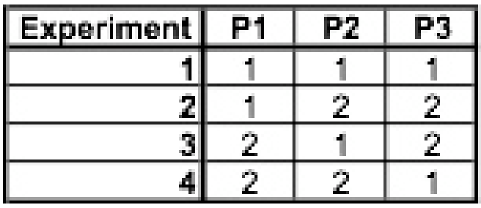
After knowing the number of variables and the number of levels, the proper orthogonal array can be selected using the array selector table shown below (Table 2.1). For example, if we have three variables and two levels for each variable, it can be seen the proper array is L4 (Table 2.2).

Table 2.1 Table selector of orthogonal array [Web Resource 1]



		Number of Parameters (P)										
		2	3	4	5	6	7	8	9	10	11	12
Number of Levels	2	L4	L4	L8	L8	L8	L8	L12	L12	L12	L12	L16
	3	L9	L9	L9	L18	L18	L18	L18	L27	L27	L27	L27
	4	L'16	L'16	L'16	L'16	L'32	L'32	L'32	L'32	L'32		
	5	L25	L25	L25	L25	L25	L50	L50	L50	L50	L50	L50

Table 2.2 L4 table [Web Resource 2]



Experiment	P1	P2	P3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

These arrays were created using an algorithm Taguchi developed, and allow each variable to be tested equally to acquire the response of the experiments. In Table 2.2, the first column is the experiment number. The second to the fourth column show the levels to be assigned to each variable (P1, P2, and P3). The number 1 and 2 can indicate the level of each variable in the experiment. From Table 2.2, we can see that if we choose any two columns (variables), all combination of the two variables are performed in the table. If we want to see the effect on response while P1 at different levels. We have to calculate the difference between the response at (P1 level = 1) and the response at (P1 level = 2). We can just sum up the response at (P1 level = 1) and (P1 level = 2), and then subtract one by the other. The result is exact the effect on response from changing the value of P1. The reason why we can neglect P2 and P3 in this case is because in experiment 1 and 2 (P1 level = 1), the two (all) levels of P2 and P3 occur, in experiment 3 and 4, the two (all) levels of P2 and P3 occur. So the contribution to response from P2 and P3 are eliminated after the subtraction.

In addition to calculate the effect of a variable on response, OA can offer many other kind of help in an experiment. In this thesis, we use OA to extinguish the variables which affect yield the most, and to provide the initial combination of all variables. The detail will be referred in chapter 3.

2.4 Analog Circuit Optimization

Automation of analog circuit design is a field of growing interest in recent decades. Nowadays, VLSI technology progresses towards the integration of mixed analog–digital circuits as a complete system-on-a-chip. Though the analog part is a small fraction of the entire circuit, it is much more difficult to design due to its complexity. Without an automated synthesis methodology, analog circuit design suffers from long design time, high complexity, and high cost [Liu 09]. Thus, automation methodologies of analog circuits get more and more important with the growing complexity of the circuits. The techniques used in analog circuit optimization can be classified into simulation-based methods and equation-based methods [Graeb 01]. Equation-based methods need to derive the design equations by symbolic analysis. Simulation-based methods use simulation to obtain the required information of the circuit. Examples of simulation-based methods are *genetic algorithms* (GA), *simulated annealing* (SA), *response surface methodology* (RSM), etc. Examples of equation-based methods are *integer linear programming* (ILP), *geometric programming* (GP), etc. Simulation-based methods have more accuracy but long run time. In the following subsections, Four methodologies of analog circuit optimization are introduced. Subsection 2.4.1 introduces geometric programming, which is equation-based. Subsection 2.4.2 introduces simulated annealing. Subsection 2.4.3 discusses genetic algorithm. Subsection 2.4.4 introduces response surface methodology. The last three techniques are simulation-based.

2.4.1 Geometric Programming

Geometric programming (GP) originated in [Zener 61] with Zener's discovery of an ingenious method for an engineer who face with the problem of optimizing a design to obtain a minimization of total operating costs [Zener 61]. Zener proposed that the operating cost can be expressed in terms of the design variables via a certain type of generalized polynomial. Unlike other analytical methods, which require the solution of nonlinear equations, GP requires the solution of a system of linear equations. Unlike numerical methods, which minimize the cost often by Newton-Ralphson method, GP provides the information of relation between the cost and associated design parameters. In [Duffin 64] and [Duffin 66], Duffin extended this method to the minimization of polynomials subject to inequality constraints on other polynomials. This development provided a nonlinear generalization to GP.

GP is a type of mathematical optimization problem. GP has the advantages that it is extremely efficient and has global optimum. The disadvantages of GP are that GP has less flexibility in the types of constraints that user can handle and the types of circuit models user can employ [Boyd 07]. In [Hershenson 99], they claimed that GP can determine whether the problem is infeasible, also, the starting point for the optimization algorithm does not have any effect on the final solution. That is, a starting point or initial design is unnecessary.

A GP is an optimization problem of the form

$$\begin{aligned} &\text{Minimize} \quad f_0(x) \\ &\text{Subject to} \quad f_i(x) \le 1,\ i = 1,\ldots,m, \\ &\qquad\qquad\quad g_i(x) = 1,\ i = 1,\ldots,p, \end{aligned} \tag{3}$$

where f_i are posynomial functions, g_i are monomial functions, and x_i are the optimization variables. There is an implicit constraint that the variables are positive, i.e., $x_i > 0$. [Boyd 07] We refer to equation (4) as a geometric program in standard form. In the standard form of GP, the objective function must be posynomial and it must be minimized; the equality constraints can only have the form of a monomial equal to 1; and the inequality constraints can only have the form of a posynomial less than or equal to 1. A GP can be reformulated as a convex optimization problem, that is, a problem of minimizing a convex function subject to convex inequality constraints and linear equality constraints. The reformulation is the key to obtain a global optimum and efficient solve geometric programs. To obtain the convex form, we have to define new variables $y_i = \log x_i$, and convert the standard form into a convex form as

$$\begin{aligned} &\text{Minimize} \quad \log f_0(e^y) \\ &\text{Subject to} \quad \log f_i(e^y) \le 0,\ i = 1,\ldots,m, \end{aligned} \tag{4}$$

$$log\ g_i(e^y) = 0,\ i = 1,\ldots,p,$$

This is so-called convex form of the geometric program. To solve the GP problem, one can find existing tool that handle objectives and constraint functions, such as MATLAB, LANCELOT, MINOS, LOQO, and LINGO-NL, etc.

In [Cheng 12], GP is applied to optimize a two-stage operational amplifier and a folded-cascode operational amplifier by sizing the transistor size. They construct the equations and apply the non-linear programming solver in MATLAB.

2.4.2 Simulated Annealing

Simulated annealing (SA) is a generic optimization algorithm for combinatorial optimization problems [Kirkpatrick 83]. At the beginning, the current state and the current temperature are set to the initial state and the initial temperature, respectively. The main loop contains the following steps. A new state is generated by randomly perturbing the current state. The cost of the new state is computed and compared to the cost of the current state. If the cost of the new state is smaller than the cost of the current state, then accept the new state. However, if the cost of the new state is greater than the cost of the current state, the probability of accepting the new state is determined by the a function of the difference of the two costs and the temperature. The probability to accept a state with a greater cost is close to one when the temperature is high, and the probability decreases exponentially as the temperature decreases. When the temperature is low, the probability to accept a state with a larger cost is close to zero. Simulated annealing provides the uphill moves mechanism so that we can escape from being trapped in the local minimum. The temperature is multiplied by a cooling factor at the end of the main loop. The main loop continues until the temperature is freezing, which means the temperature is below a specified level.

There are four key ingredients for simulated annealing. They are solution space, neighborhood structure, cost function, and annealing schedule. Solution space defines the region of the feasible solution. Neighborhood structure indicates how to find a neighboring solution from the current state. Cost function set the rule to evaluate the quality of a solution. Annealing schedule shows how to conduct the search process to find a desired solution. Figure 2.13 shows the pseudo-code of simulated annealing algorithm.

```
Get an initial state $S$;
Set an initial temperature $T > 0$;
while not “frozen” do
    new state $S'$ = a random new neighbor from $S$
    $delta\_cost = cost(S') - cost(S)$;
    if $delta\_cost \leq 0$ then $S = S'$
    if $delta\_cost > 0$ then $S = S'$ with probability $e^{-delta\_cost / T}$;
    $T = rT$;     // cooling mechanism
return $S$
```

Figure 2.13 Simulated annealing algorithm

SA is a simulation-based approach for circuit optimization. After every perturbation, users have to evaluate the circuit performance before the next perturbation. Hence, although SA can provide more accurate optimization results, it often takes too much time. In [Gielen 90], the authors use analytic circuit models accompanied with SA to size all circuit elements in order to satisfy the performance constraints of an analog circuit. The form the design space for SA based on the analytic circuit models and the design space consists of the points which satisfy the design constraints. In the initialization of the SA algorithm in [Gielen 90], the independent variables are gridded over their initial range. They claim this limitation to the range can reduce the CPU time.

2.4.3 Genetic Algorithm

Genetic algorithm (GA) is a search heuristic that imitates the process of natural evolution. This heuristic is often used to generate useful solutions to optimization problems and has been widely used in analog *computer-aided design* (CAD) for real-valued optimization [Smedt 03][Tiwary 06]. Moreover, GA can be applied in many different realms, such as bioinformatics, economics, chemistry, manufacturing, etc. and can be used to solve traveling salesman problem, combinatorial optimization, and structural synthesis of circuits [Aggarwal 07].

GA is based on the Darwinian principle of natural selection and operates on the principle of “survival of the fittest”, generating new design solutions from a population of existing solutions and disgarding the design solutions which have an inferior performance or fitness [Grimbleby 00]. GA is inherently robust and has been shown to efficiently search large solution spaces containing discrete or discontinuous parameters and non-linear constraints, without being trapped in local minima. Besides, GA does not require initial guess of parameters [Noren 01].

Then core of GA is the concept of chromosome. The chromosome contains all information necessary to describe an individual. Each member of the population has a “chromosome” which consists of a number of “genes”. Each gene represents one part of the design solution. The flow of GA used for analog circuit design usually follow the general GA flow. In analog circuit design, the purpose of GA is to determine the elements of the unknown vector (chromosome) to maximize the fitness function [Taherzadeh-Sani 03]. The fitness function is a function which determines the fitness of the solution subjected to the design objectives. The fitness function can be constructed using any output variable available from a simulator (such as SPICE). The traditional flow of GA is shown in Figure 2.14.

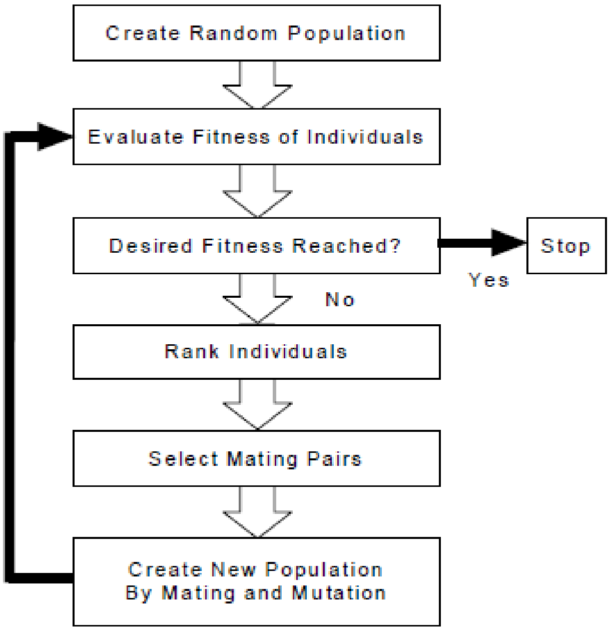


Figure 2.14 The flow of GA [Noren 01]

The first step is to create the first generation population. The population size depends on the problem size. The population is usually generated randomly. Populating the first generation with known circuits has the advantage that it may lead to a faster convergence, but at the same time, it may limit the GA from exploring the designs that designer may ignore [Fujii 01]. The second step is to evaluate the fitness of individuals. In this step, user should first decode the population of chromosomes into a format that is recognizable by a circuit simulator (such as SPICE). The circuit simulator will simulate each solution and then the fitness function is applied on the output data to recognize if the individual meets the predefined design objectives and constraints.

Then the individuals are ranked according to fitness. Individuals are selected for mating based on fitness. Individuals with higher fitness have higher probability of mating and passing on genetic information to subsequent generations while individuals with lower fitness have a non-zero probability of mating to preserve diversity. Mating is simulated by applying the crossover operation to the chromosome of parents. Mutation is simulated by randomly changing a few bits in the chromosome of the offspring. Mutation can provide a mechanism for exploring new regions of the solution space and prevent early convergence to local minima. Finally, the fitness of the new generation is evaluated and the process is repeated for a number of generations or until a desired fitness is achieved.

2.4.4 Response Surface Methodology

Response surface methodology (RSM) is a collection of statistical techniques which are useful for optimization [Myers 02]. RSM was introduced by G. Box and K. Wilson in 1951 [Box 51][Hill 66]. RSM allows user to see alternative conditions as well as the sensitivity and the roles of the design variables in the design space [Myers 91]. RSM is applied in situations where several input variables influence the performance measure of the process. The performance measure is called the *response*. The field of RSM consists of 1) experimental strategy for exploring the space of the process or independent variables, 2) empirical statistical modeling to develop an appropriate approximating relationship between the response and the input variables, and 3) optimization methods for finding the values of the input variables that produce desirable values of the response. In general, we can write down the relationship between response (yield, in this thesis) and input variables as:

$$y = f(\xi_1, \xi_2, ..., \xi_\kappa) + \varepsilon \tag{5}$$

, where ε is a term that represents other sources of variability not accounted for *f*, that is, noise. ξ is the input variables. y is the response. Usually ε is treated as a statistical error, often assuming it to have a normal distribution with mean zero and variance σ^2. The surface describing the relationship between input variables and response is called response surface. It is just like we use a function to model the relationship between inputs and outputs of an experiment. Figure 2.15 shows a response surface and contour plot of an experiment with input variable x, y, and response z for example.

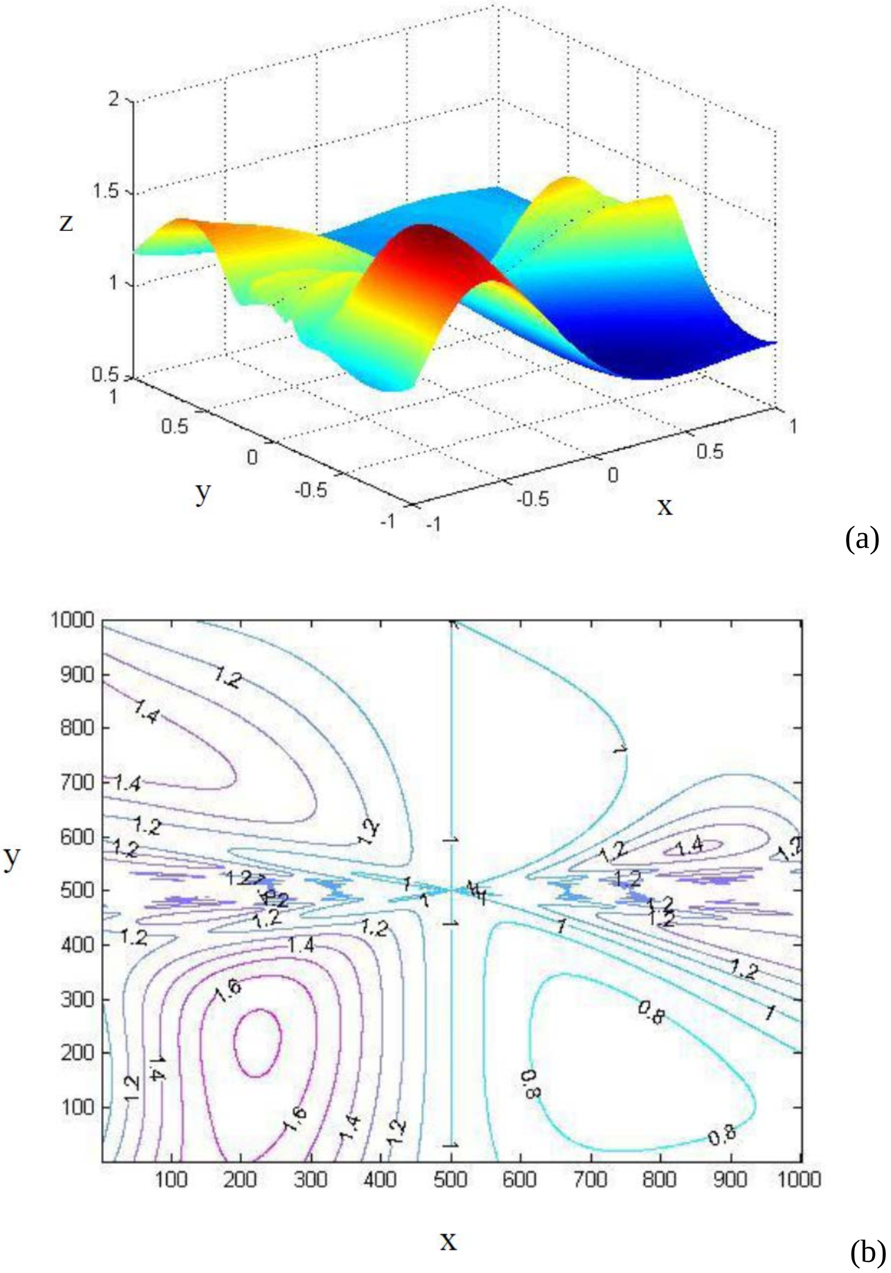


Figure 2.15 (a) response surface and (b) contour plot of an experiment with input variable x, y, and response z

In RSM, we can use a first-order polynomial model or second-order polynomial model to approximate the response surface of the experiment. However, the models are only approximations. With the advantages that the models are easy to estimate and apply, RSM still use it with mechanisms that ensure the adequacy of the model.

RSM is a sequential experimental process that includes three steps. The first step is screening experiment. The goal of this step is to indicate the principle factors that influence the response more and eliminate unimportant design variables, so that reduce the number of input variables. It is the preliminary step to make the subsequent experiments more efficient. The second step is the steepest ascent (descent) method on first-order response surface models to optimize the process. The goal of this step is to move the response along the path toward the optimum by adjusting the values of input variables. If the current settings of the input variables are not consistent with optimum performance, the experimenter should determine a set of adjustments to the input variables that will move the process toward the optimum. The path of steepest ascent is computed in the expectation that the maximum increase in response. Steepest descent produces a path that results in a maximum decrease in response. When the process is near the optimum, we begin the third step. At this point, experimenter usually wants a model that will accurately approximate the true response function within a relatively small region around the optimum, a second-order model then be used. Once an approximating model has been obtained, this model may be analyzed to determine the optimum conditions for the experiment [Carley 04].

RSM is widely used in analog circuit optimization. For example, in [Su 93], the authors used RSM to optimize the area of a Miller compensated operational amplifier. In [Graeb 01], RSM is used to do analog circuit sizing. RSM serves to replace computationally expensive circuit simulation models with cheaper performance evaluation by analytical functions. In [Alvarez 88], RSM is used to simultaneously determine an optimal operating point and analyze its sensitivity to process and device perturbations. The approach of RSM and how they apply RSM to computer-aided VLSI device design is introduced in detail in [Alvarez 88].

2.5 FlexiAnalyzer

FlexiAnalyzer is a SPICE-based simulator proposed by [Ma 11]. FlexiAnalyzer supports four types of analysis – 1) aging analysis, which simulates the circuit and generates aged TFT models with threshold voltage shifted to observe the performance of TFT device after a certain operation time. 2) performance analysis, which simulates the aged and/or bent circuits for measuring performance degradation. 3) yield analysis, which predicts the yield of aged and/or bent circuits. 4) weak-spot analysis, which simulates the circuit with one TFT degraded at a time to identify the weakest transistor in the design. The details of the four types of analysis are discussed below respectively.

The aging analysis provides an aged TFT model for every transistor in the circuit. Figure 2.16 shows the flow of aging analysis. This analysis requires a circuit netlist, input stimuli, and a new (not-used) TFT model as inputs. A transient simulation is performed over a short period of time. To save the simulation time, V_{th} is assumed to be the same during the simulation. V_{th} shift of every TFT is then calculated according to aging equations with a period of operation time.

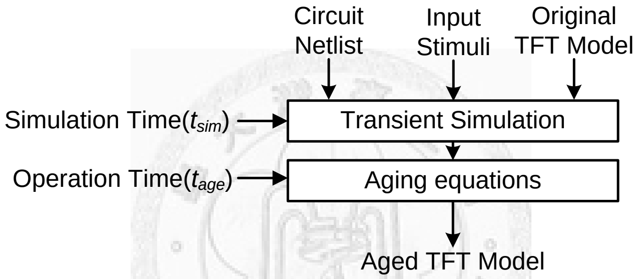


Figure 2.16 Flow of aging analysis

The performance analysis simulates the circuit with three effects considered – bending effect, aging effect, and process variation. In performance analysis, three parameters have to be specified: change in mobility, operation time of device under test, and the original threshold voltage. This analysis helps designers to predict the performance of the circuit under different conditions.

The yield analysis applies *Monte Carlo* simulations to estimate the yield given a specific process variation. FlexiAnalyzer uses the *stratified sampling* method, in which the initial threshold voltages are partitioned into several intervals and are randomly generated according to the probability of each interval. Figure 2.17 shows an example distribution of initial threshold voltage. The distribution is partitioned into 10 intervals. Considering the bending and aging effects, yield analysis supports four modes: *new/flat*, *new/bent*, *aged/flat*, and *aged/bent mode*. In new/flat mode, circuits are simulated without bending effect and aging effect (mobility and threshold voltage remain unchanged). In new/bent mode, the circuit is simulated with mobility change in new condition (operation time is 0). In aged/flat mode, the circuit is simulated without bending after a specified operation time. In aged/bent mode, the circuit is simulated with specified mobility change after a specified operation time.

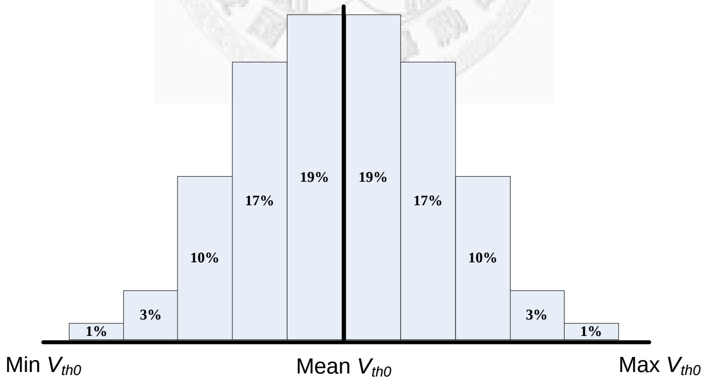


Figure 2.17 Example distribution of V_{th0}

The weak-spot analysis simulates the circuit with one TFT changed at a time to estimate the sensitivity of the output to each transistor in the circuit. Figure 2.18

shows the block diagram of weak-spot analysis. In the loop, one TFT is changed and simulated for the corresponding output change. Then the TFT is restored and another TFT is changed. This iteration is repeated until every TFT in the circuit has been simulated. Weak-spot analysis supports three modes: *bending, aging, and process variation mode*. In bending mode, mobility of a single TFT is changed to a specified value while threshold voltage fixed. In aging mode, threshold voltage of a single TFT is shifted according to the specified operation time while mobility fixed. In process variation mode, initial threshold voltage of a single TFT is shifted to the specified value. After the simulation, weak-spot analysis will rank the TFT according to the sensitivity.

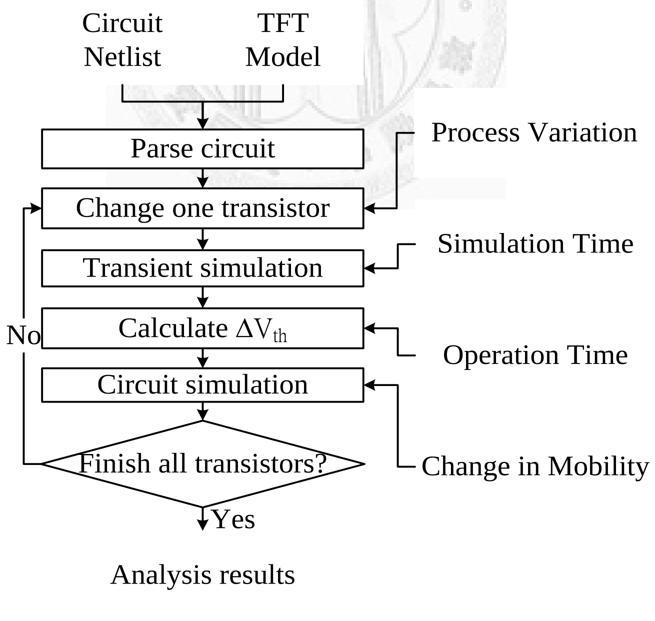


Figure 2.18 Flow of weak-spot analysis

Chapter 3 Proposed Technique

Chapter 3 introduces the proposed technique of this thesis. Section 3.1 introduces the overall flow of our tool, *FlexiOptimizer*. Section 3.2 discusses how the tool calculates the yield considering the *process variation*, *aging effect*, and *bending effect*. Section 3.3 introduces the details of the RSM flow for *screening experiment* and *yield optimization*.

3.1 FlexiOptimizer

In this thesis, a yield optimizer for flexible TFT circuits – *FlexiOptimizer,* is proposed. Figure 3.1 shows the overall flow of the proposed tool. The flow contains two main parts, 1) a SPICE-based yield simulator – *YieldAnalyzer* that considers three important effects on flexible TFT analog circuit. The three effects are V_{th} variation, aging effect, and bending effect. Users have to assign the V_{th} distribution, operation time, and mobility range as inputs. 2) RSM, an optimization technique which performs screening experiment to distinguish important variables and then do yield optimization.

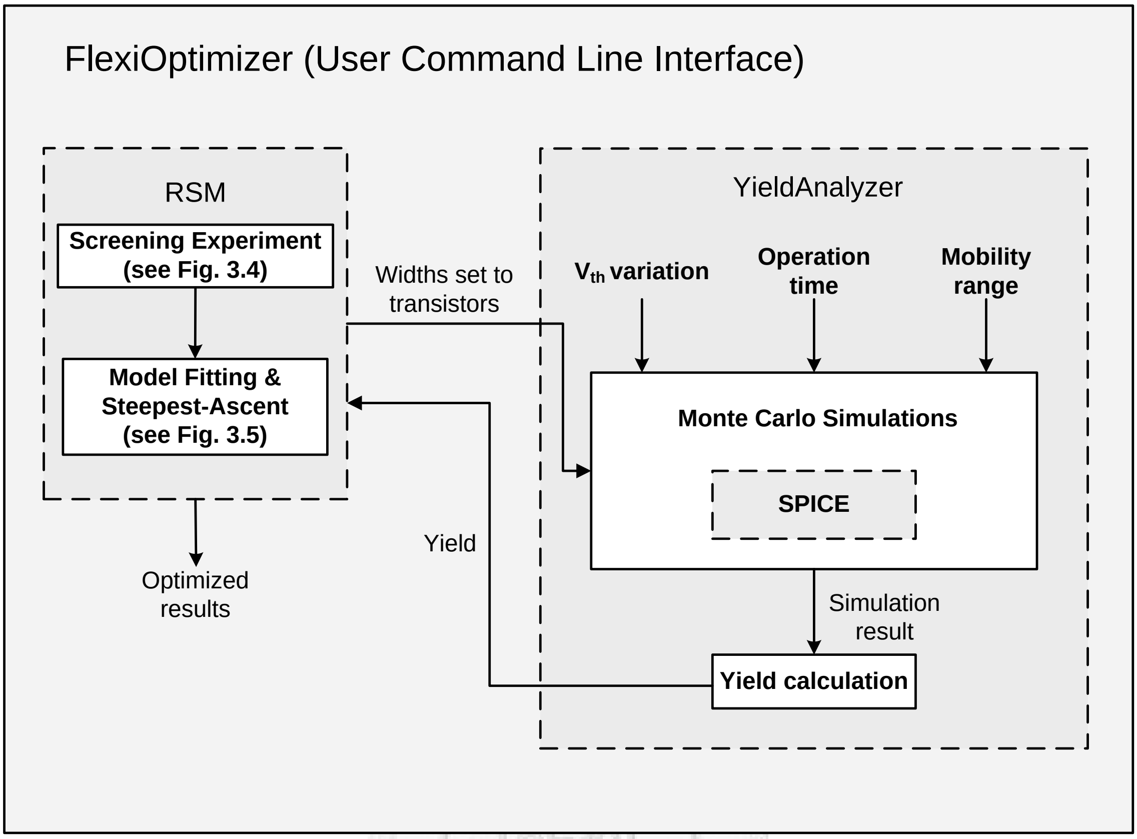


Figure 3.1 Overall flow of FlexiOptimizer

In FlexiOptimizer, the first step is to parse the circuit with a spice file that describes the circuit connection, input waveform, and TFT model. After all the settings prepared, RSM begins *screening experiment* to eliminate unimportant variables so that less variables have to be considered in yield optimization step. During the RSM, RSM will repeatedly generate the combination of the variables (widths of transistors) based on the point selected by RSM and set the values to SPICE, and then request YieldAnalyzer to report the yield under the condition. When YieldAnalyzer is called, it will perform SPICE simulations considering aging effect, bending effect, and V_{th}

variation and calculate the yield under current condition, and then return the yield value to RSM. The detail of yield optimization step is discussed in section 3.3. After yield optimization step, optimized result and the width of each transistor with the best yield will be reported.

3.2 YieldAnalyzer

YieldAnalyzer is based on the yield-analysis of FlexiAnalyzer [Ma 11]. In YieldAnalyzer, three effects are considered when calculating the yield. They are process variation of initial V_{th} (V_{th0}), aging effect, and bending effect. In this section, the three effects are discussed respectively.

3.2.1 Process Variation of $\mathbf{V_{th0}}$

When flexible TFT circuits are fabricated, process variation occurs and results in a statistical spread on the circuit component parameter value. Process variation of V_{th} causes measurable and predictable variance in the output performance of all circuits and particularly analog circuits due to mismatch. Thus, process variation may reduce the circuit yield and should be considered in a yield analyzer.

In the proposed tool, we considers two process variation types – *inter-die variation* and *intra-die variation*. We use stratified sampling [Pengelly 02] to consider the inter-die variation of V_{th0}. Figure 3.2 shows the inter-die variation. User can define the *maximum* V_{th0}, the *minimum* V_{th0}, *number of subintervals*, *percentage value of each subinterval*, and *number of Monte Carlo trials*. The range of each subinterval is

$$\frac{max\ V_{th0} - min\ V_{th0}}{number\ of\ subintervals} \tag{7}$$

The percentage of each interval is 2%, 4%, 10%, 16%, 18%, 18%, 16%, 10%, 4% and 2% of total circuits (number of Monte Carlo trials). Given the distribution, the V_{th0} of circuit in the interval is randomly assigned between the upper bound and the lower bound of the subinterval.

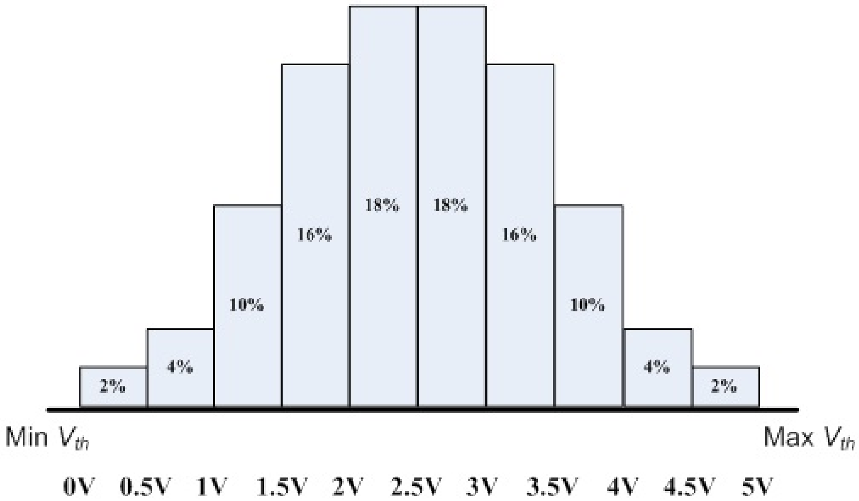


Figure 3.2 Example of inter-die variation

We consider the intra-die variation by normal distribution. User only have to assign the *sigma* (standard deviation) of the normal distribution, and we will add the ΔV_{th} caused by intra-die variation on the V_{th} generated by the inter-die variation mentioned above. The initial threshold voltage (V_{th0}) of each circuit then obtained by adding inter-die variation and intra-die variation, as the following equation,

$$V_{th0} = V_{th,\mathrm{int}\,er-die} + V_{th,\mathrm{int}\,ra-die} \tag{8}$$

Finally, we have all V_{th0} of each transistor of all circuits (number of circuits = Monte Carlo trials).

3.2.2 Aging Effect

Aging effect on flexible TFTs was discussed in section 2.2.2. We model the aging effect as V_{th} degradation. In proposed tool, we quantitatively predict the degradation with two models for both a-Si technology and IGZO technology. The aging model for a-Si TFT is proposed by [Shringarpure 07]. An intermediate parameter called *Age* is proposed. The parameter *Age* is used to quantify the degradation in individual TFT in the circuit and is related to ΔV_{th} as

$$\Delta Age(\Delta t_i) = \int_{t_{i-1}}^{t_i} A^{-\beta} \exp(\frac{-E_A}{\beta kT})(V_{GS} - \eta V_{DS} - V_{th})^{\frac{n}{\beta}} dt \tag{8}$$

$$\eta = \frac{V_{GS}}{V_{GS} + V_{DS}} \tag{9}$$

Where k is Boltzmann constant, T is the absolute temperature, t is the bias stress time duration, E_A is the mean activation energy, A is the degradation rate, β and n are process related constants. V_{GS} and V_{DS} are assumed to be constant during $\triangle t_i = t_i - t_{i-1}$. The incremental *Ages* (*△Age*) are summed to obtain the total *Age* ($Age(t_{total})$) at the end of simulation time t_{total}. Finally, the ΔV_{th} caused by aging effect after an operation time t_{total} is obtained by the equation:

$$\Delta V_{th}(t_{total}) = [Age(t_{total})]^{\beta} \tag{10}$$

In our tool, the parameter used is fitted in [Shringarpure 07], where $A^{-\beta}exp(-E_A/kT)$ *is* 0.025, n is 1.0, and β is 0.25 for a-Si TFT technology.

[To be filled] The aging model for IGZO TFT used in our work is based on the stretch-exponential model [Libsch 93]

$$\left|\Delta V_{th}\right| = \left|\Delta V_0\right|\{1 - \exp[-(\frac{t_{stress}}{\tau})^{\beta}]\} \tag{11}$$

where $\varDelta V_0$ is the effective voltage drop across the gate insulator, β is the stretched exponential exponent, τ represents the characteristic trapping time of carriers. The numbers we use in this paper are: β=0.38 and τ=1,260,000 seconds [Su 10].

In proposed tool, we set four parameters: *total operation time* (t_{total}), *interval time* ($t_{interval}$), *number of simulation with same* $t_{interval}$ ($Num_{interval}$), and *amplification ratio* (*ratio*). At the beginning of each $t_{interval}$, we run DC simulation to obtain each node's bias. We give an input signal, assuming the circuit is continually used. We obtain each node's bias and use equation (8) to calculate each transistor's ΔAge parameter during the current $t_{interval}$. After a $t_{interval}$, the bias of each node will be different. We run another DC simulation with new bias and obtain the ΔAge during the current $t_{interval}$. This step will be repeated until all time steps are done. Figure 3.3 shows the detail of calculating *Age* and V_{th}. The dark red line of the figure is the possible *Age* model, the blocks show how we approximate the *Age* model. After each block, we add $t_{interval}$ to t_{total}. We use a *counter* to keep track of the number of simulations. If the *counter* is equal to $Num_{interval}$, then we multiply it by *amplification ratio* to obtain new $t_{interval}$ and set *counter* to 0. Finally, when the t_{total} is reached, the final ΔV_{th} is obtained.

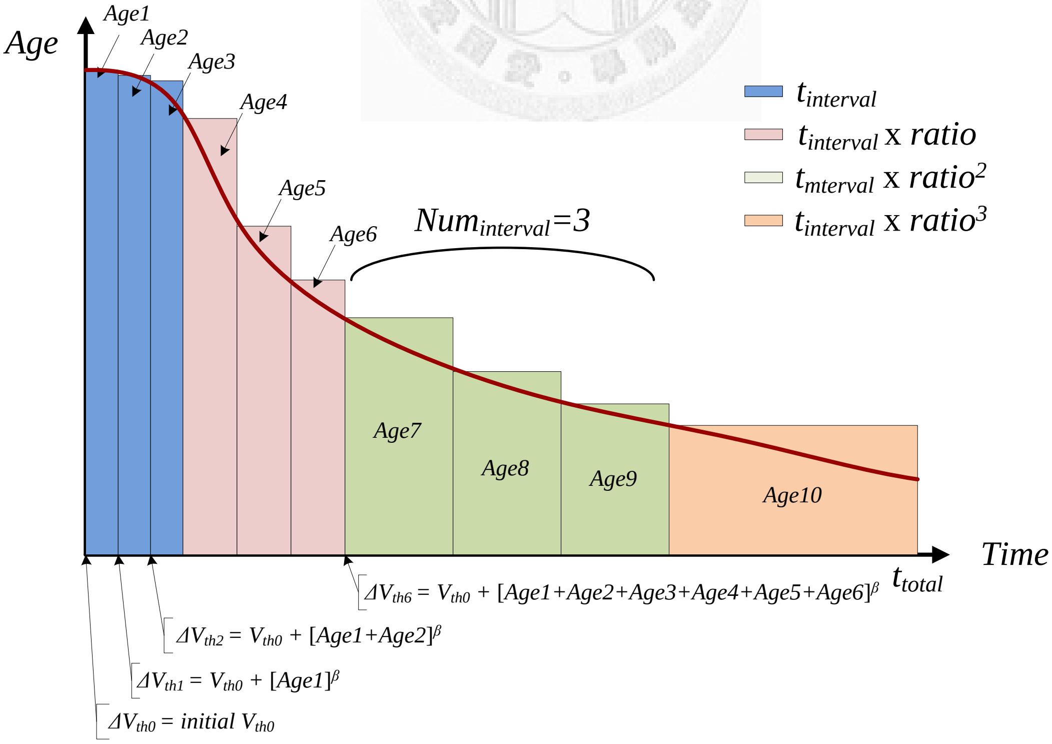

Age
Age1
Age2
Age3
Age4
Age5
Age6
Age7
Age8
Age9
Age10
$Num_{interval}=3$
$t_{interval}$
$t_{interval}$ x $ratio$
$t_{mterval}$ x $ratio^2$
$t_{interval}$ x $ratio^3$
Time
t_{total}
$\Delta V_{th6} = V_{th0} + [Age1+Age2+Age3+Age4+Age5+Age6]^\beta$
$\Delta V_{th2} = V_{th0} + [Age1+Age2]^\beta$
$\Delta V_{th1} = V_{th0} + [Age1]^\beta$
$\Delta V_{th0} = initial\ V_{th0}$

Figure 3.3 Detail of calculating Age and V_{th}

3.2.3 Bending Effect

When TFT circuits are bent, mobility of transistors may change. Thus, the yield of TFT circuits reduces. Therefore, the proposed tool takes bending effect into consideration. Table 3.1 shows the percentage of change in mobility of a-Si TFT and IGZO TFT technologies. The range of strains ε (where $\varepsilon = \Delta L/L_0$, with L_0 = initial gate length and ΔL = the change of gate length) are listed in the table. If ε exceeds the range, the TFT will be damaged permanently.

Table 3.1 Percentage of change in mobility of four TFT technologies

TFT technology	Compressive strain	Tensile strain
a-Si TFT [Gleskova 02]	-26% (ε=0%~1%)	7.6% (ε=0%~0.2%)
a-IGZO TFT [Munzenrieder 11]	-2% (ε=0%~0.27%)	3% (ε=0%~0.3%)

In the proposed tool, we consider both compressive strain and tensile strain. We set two different mobility changes to the circuit respectively. The circuit pass when it pass both cases that the mobility change under compressive strain and tensile strain. Take a-Si TFT circuits for instance, the mobility change of each transistor will be set to -26% and perform the simulation. Then the mobility change of each transistor will be set to 7.6% and perform the simulation. If the circuit pass the specification under two conditions, the circuit will be judged pass. Otherwise the circuit will be judged fail.

3.3 Response Surface Methodology

This section presents the proposed RSM flow. RSM can be divided into two main parts – 1) screening experiment, and 2) model fitting & steepest-ascent. The flow of screening experiment is shown in Figure 3.4.

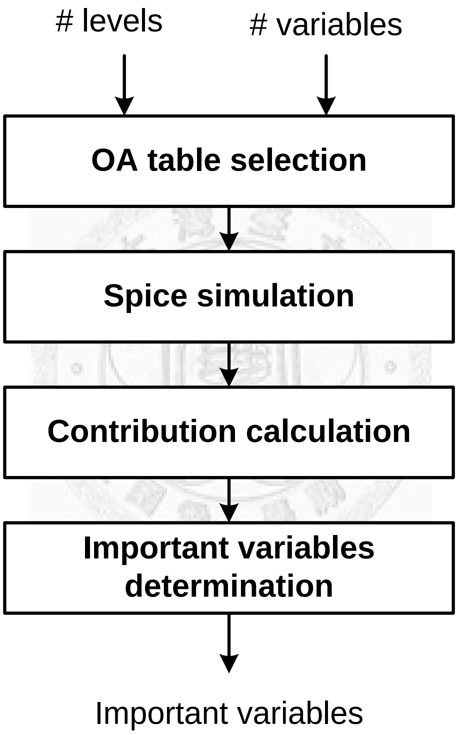


Figure 3.4 flow of screening experiment

The first step to begin the RSM flow is screening experiment. Screening experiment select several representative points to simulate and identify the important variables according to the simulation results. After screening experiment is performed, important variables are kept and used in the model fitting step. The flow of model fitting & steepest-ascent is shown in Figure 3.5.

In RSM, we define four parameters. 1) *RSM point* is an assignment of k important variables with minimum size assigned to unimportant variables. An RSM point can be written in a vector $\boldsymbol{X} = [x_1, x_2, \ldots x_k]^T$. 2) *Design center* $\boldsymbol{X}^* = [x_1^*, x_2^*, \ldots x_k^*]^T$ is the best known RSM point so far. 3) *Design space* is the solution space that satisfies two conditions. (a) within the *radius r* of the design center, and (b) satisfies the user-defined constraints. 4) *Radius r* is a scalar that represents the maximum distance between design center and the boundary of the design space. In this thesis, all TFT transistor sizes are constrained between 10 and 500 μm, so r = 245 μm at the beginning. r is iteratively shrunk by 20% reduction in each first-order model fitting & steepest ascent iteration.

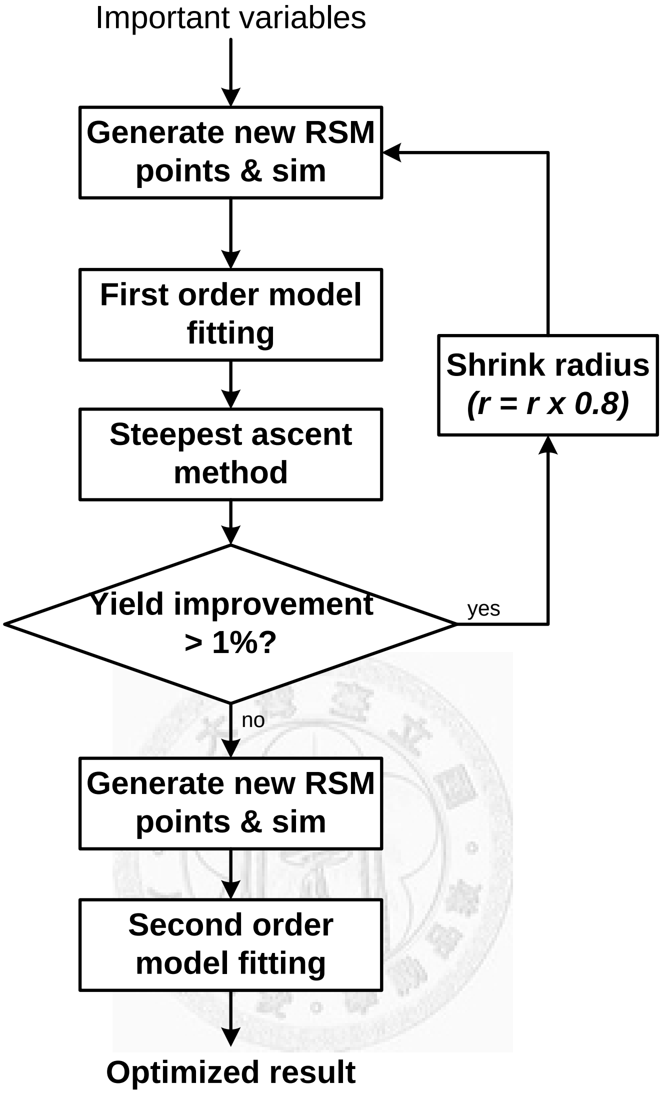

Important variables
Generate new RSM points & sim
First order model fitting
Steepest ascent method
Yield improvement > 1%?
yes
Shrink radius (r = r x 0.8)
no
Generate new RSM points & sim
Second order model fitting
Optimized result

Figure 3.5 flow of model fitting & steepest-ascent

In the model fitting & steepest-ascent part, we randomly generate *new RSM points*, which are used to fit the model, and then perform Yield Analysis on the new RSM points. After the chosen points are all simulated, we perform first-order model fitting process to obtain a first-order model which fits the response surface by a model with the highest power of each variable to be one. First-order model just approximates the trend of the response surface. Though it is not precise enough, it provides us a fast way that we can roughly find where of the design space will have the higher response.

After fitting the first-order model, we will locate the design center on the point with the highest yield currently and then perform the steepest-ascent method from the design center. Steepest-ascent method is to find the steepest direction around the design center and then move along the path to make a possibly largest ascent of the response. After first-order model fitting and steepest-ascent method are performed, the program will check if the yield are increased by more than 1%. If so, it means the process is still useful to promote the yield, so we enter the loop again, expecting that we can improve the yield further. Besides, we also expect that the current solution is closer to the best solution. Thus, we no longer need the design space to be the same size, so the size of the radius *r* will be multiplied by 0.8. That is, we shrink the design space. If the yield increases no more than 1%, it means the process may not promote the yield efficiently anymore. Thus, we leave the loop and generate new simulation points for second-order model fitting. Before entering second-order model fitting, we will check if the design space is small enough. If the design space is still large, we will shrink the design radius to its original value multiplied by 0.2. After the second-order model fitting, the optimized result is found by calculating the stationary point. The details are introduced in the subsections.

Subsection 3.3.1 describes how we implement screening experiment in the proposed tool. Subsection 3.3.2 introduces first-order model fitting steps by steps. Subsection 3.3.3 introduces how we apply the steepest ascent method and the features included. Subsection 3.3.4 introduces the process of second-order model fitting. Subsection 3.3.5 discusses how we calculate the final optimized result after we fit a second-order model. Subsection 3.3.6 discusses how we handle the correlation terms.

3.3.1 Screening Experiment

Screening experiment is used to identify the *important variables* which affect the output response of the circuit the most. After screening experiment, only important variables are left. Thus, we can reduce the effort in yield optimization. We propose an orthogonal array (OA)-based screening experiment method in our tool.

The screening experiment begins at OA table selection. OA is a suitable candidate to obtain enough information in relatively low simulation times. We choose the OA table according to the *number of variables* and the *number of levels*. An OA table consists of the assigned level to each variable in each iteration. The corresponding values of the assigned levels are determined by the designer. The length of a column in

OA table stands for the number of simulations have to be performed. Each row of an OA table represents an iteration of simulation. After obtaining the OA table, we begin the second step – perform the simulations. The assigned value to all input variables are set according to the OA table. After all the simulations are performed, we obtain the yield of each iteration, then the third step begins. The third step is to calculate the *contribution* to the yield of each variable. The *contribution* of a variable is determined by the equation:

$$contribution(A) = \frac{\sum yield(A\ at\ highest\ level)}{number\ of\ A\ at\ highest\ level} - \frac{\sum yield(A\ at\ lowest\ level)}{number\ of\ A\ at\ lowest\ level} \qquad (6)$$

If the *number of levels* is 2, and the levels are high and low. The *contribution* of variable A is determined by the average yield in simulations when the level of A is high minus the average yield in simulations when the level of A is low. After the calculation of *contribution*, each variable has its corresponding *contribution* in percentage. The higher the *contribution*, the more the variable affects yield. Finally, we keep the variables with *contribution* more than 5% as important variables. If there is only one variable satisfied, we will choose two variables with the highest two *contribution* values to be important variables. After performing the screening experiment, the chosen important variables will be fed into the yield optimizer.

3.3.2 First-Order Model Fitting

As mentioned before, the true response of an experiment is unknown. Therefore, we can only approximate the true response by an approximating model. In the early stage of RSM flow, the model used to approximate the true response is first-order model. To build a first-order model, one should perform several simulations, and then fit the first-order model based on the simulation points. This is called *first-order model fitting*. In this subsection, the process of fitting a first-order model is shown in detail.

Suppose that we have *k* important variables in the model fitting. A first-order response surface model which decribes the relationship between input variables and output response is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon \tag{11}$$

where y represents the output response, x_1 to x_k represent the important variables, β_1 to β_k are called the regression coefficients, and ε is noise. We treat ε as a statistical error, and often assuming it to have normal distribution with mean zero and variance σ^2. Suppose now there are *n* (>*k*) experiments available. We may write the equation (11) in terms of the experiments as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + \varepsilon_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \qquad (12)$$

where *i* denote the experiment number, x_{ij} denote the *i*th experiment of variable x_j. The method of least squares chooses the β's in equation (12) so that the sum of the squares of the errors ε_i are minimized. The least squares function *L* is

$$L = \sum_{i=1}^{n} {\varepsilon_i}^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2 \qquad (13)$$

The function *L* is to be minimized with respect to $\beta_0, \beta_1, ..., \beta_k$. The least squares estimators, $b_0, b_1, ..., b_k$, must satisfy

$$\left.\frac{\partial L}{\partial \beta_0}\right|_{b_0, b_1, ..., b_k} = -2\sum_{i=1}^{n} (y_i - b_0 - \sum_{j=1}^{k} b_j x_{ij}) = 0 \qquad (14)$$

and

$$\left.\frac{\partial L}{\partial \beta_j}\right|_{b_0, b_1, ..., b_k} = -2\sum_{i=1}^{n} (y_i - b_0 - \sum_{j=1}^{k} b_j x_{ij}) x_{ij} = 0, \qquad j = 1,2,...,k \qquad (15)$$

Simplifying equation (14) and (15), we can obtain

$$nb_0 + b_1\sum_{i=1}^{n} x_{i1} + b_2\sum_{i=1}^{n} x_{i2} + \ldots + b_k\sum_{i=1}^{n} x_{ik} = \sum_{i=1}^{n} y_i$$

$$b_0\sum_{i=1}^{n} x_{i1} + b_1\sum_{i=1}^{n} x_{i1}{}^2 + b_2\sum_{i=1}^{n} x_{i1}x_{i2} + \ldots + b_k\sum_{i=1}^{n} x_{i1}x_{ik} = \sum_{i=1}^{n} x_{i1}y_i$$

$$\cdots \tag{16}$$

$$b_0\sum_{i=1}^{n} x_{ik} + b_1\sum_{i=1}^{n} x_{ik}x_{i1} + b_2\sum_{i=1}^{n} x_{ik}x_{i2} + \ldots + b_k\sum_{i=1}^{n} x_{ik}{}^2 = \sum_{i=1}^{n} x_{ik}y_i$$

Equation (16) is called the *least squares normal equations*. It is simpler to solve the equations in matrix notation. We may write the equation in matrix notation as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{17}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \text{ and } \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

We want to find the vector of least squares estimators, **b**, which minimizes

$$L = \sum_{i=1}^{n} {\varepsilon_i}^2 = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{y}^{\mathbf{T}}\mathbf{y} \text{ - } \mathbf{2}\boldsymbol{\beta}^{\mathbf{T}}\mathbf{X}^T\mathbf{y} + \boldsymbol{\beta}^{\mathbf{T}}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \tag{18}$$

since $\boldsymbol{\beta}^{\mathbf{T}}\mathbf{X}^T\mathbf{y}$ is a scalar and its transpose $\boldsymbol{\beta}^{\mathbf{T}}\mathbf{X}^T\mathbf{y} = \mathbf{y}^{\mathbf{T}}\mathbf{X}\boldsymbol{\beta}$ is a scalar, too. Thus,

$$\frac{\partial L}{\partial \boldsymbol{\beta}}\bigg|_{\mathbf{b}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{b} = 0 \tag{19}$$

which simplifies to

$$\mathbf{X}^T\mathbf{X}\mathbf{b} = \mathbf{X}^{\mathbf{T}}\mathbf{y} \tag{20}$$

multiply both sides by $\mathbf{X}^T\mathbf{X}$ we have

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{21}$$

The vector **b** is the coefficients of each variable in first-order model.

3.3.3 Steepest-Ascent Method

Steepest-ascent method is used in searching for a region of improved response and maximizing the response. The steepest-ascent method is applied in our tool after the first-order model fitting. After the first-order model fitting, the design center is set to be the point with the largest yield, and then steepest-ascent method begins at the design center. Figure 3.6 shows the pseudo-code of our steepest-ascent method. At first, steepest-ascent calculates the steepest path around the design center. To express the steepest direction, we define a step vector $\boldsymbol{D} = [\Delta x_1, \Delta x_2, \ldots, \Delta x_k]^T$ to represent one step in steepest-ascent. Starting from the design center, we perform Yield Analysis on RSM points within the design space, incrementally moving one step at a time. We also perform one more yield analysis on the RSM point on the design space boundary. We choose the RSM point of the highest yield as the new design center. One feature in our tool is worth mentioned. Generally, if a step just cross the boundary of the design space, steepest-ascent is stopped and the first point outside the design space will be discarded. In our tool, we scale the step back and set the end of this step to be right on the boundary of design space. Thus, we preserve the possibility that the maximum yield may occur right on the boundary.

```
// given design center, $\boldsymbol{X}^* = [x_1^*, x_2^*, \ldots, x_k^*]^T$
// given yield of $\boldsymbol{X}^* = y^*$
// given radius $r$
SA ($\boldsymbol{X}^*, y^*, r$)
    calculate direction vector $\boldsymbol{D}$
    while ($\boldsymbol{X} + \boldsymbol{D}$ still in design space)
        $\boldsymbol{X} = \boldsymbol{X} + \boldsymbol{D};$
        $yield$ = yield analysis ($\boldsymbol{X}$);
        if ($yield > y^*$)
            $\boldsymbol{X}^* = \boldsymbol{X};$    $y^* = yield;$
    calculate $t$ such that $\boldsymbol{X} = \boldsymbol{X} + t\boldsymbol{D}$ on boundary of design space
    $yield$ = yield analysis ($\boldsymbol{X}$);
    if ($yield > y^*$)
        $\boldsymbol{X}^* = \boldsymbol{X};$    $y^* = yield;$
return ($\boldsymbol{X}^*, y^*$);
```

Figure 3.6 Pseudo-code of steepest-ascent method

The movement in variable x_j (j=1,2,…,k) along the path of steepest ascent is proportional to the magnitude of the coefficient b_j with the direction taken being the sign of the coefficient. For example, if the fitted first-order model is y=5-0.5x_1 +x_2, the path of the steepest ascent will result in x_1 moving in a positive direction and x_2 moving in a negative direction. In addition, x_1 will move half as fast. Considering a fitted first-order model

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_k x_k$$

By calculating ***D***, we are producing a maximum estimated response with the constraint that $\sum_{i=1}^{k} x_i^{\,2} = \rho^2$. That is, we search for the point for which y is maximized from all points that are distance ρ from the design center. The solution to this problem takes the use of Lagrange multipliers M.

$$M = b_0 + b_1 x_1 + b_2 x_2 + ... + b_k x_k - \lambda(\sum_{i=1}^{k} x_i^{\,2} - \rho^2) \tag{22}$$

The derivative with respect to x_j is

$$\frac{\partial M}{\partial x_j} = b_j - 2\lambda x_j \qquad (j=1,2,\ldots,k) \tag{23}$$

Setting $\frac{\partial M}{\partial x_j} = 0$ gives the coordinate of x_j of the path of steepest-ascent,

$$D_j = \frac{b_j}{2\lambda} \qquad (j=1,2,\ldots,k) \tag{24}$$

Now, we let $\frac{1}{2\lambda}=q$ to be a constant of proportionality. That is, all elements of ***D*** is obtained by

$$D_1 = qb_1, D_2 = qb_2, ..., D_k = qb_k \tag{25}$$

3.3.4 Second-Order Model Fitting

The process to fit a *second-order model* is just like the first-order one. A standard form of a second-order model is shown below:

$$y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \sum_{i<j} \sum_{j=2}^{k} \beta_{ij} x_i x_j \tag{26}$$

If we treat the second-order terms x_j^2 and interaction terms $x_i x_j$ as new variables, the second-order model becomes the same form as first-order model. Then we can solve the least squares equations the same way as solving the first-order least squares equations. Thus, the coefficient vector **b** of the second-order model is the same as equation (21)

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

In our tool, we perform the matrix calculation by *GNU Scientific Library* (GSL) [Web Resource 3]. The inverse matrix operation is performed by *LU decomposition* in GSL.

3.3.5 Second-Order Model Result Calculation

After fitting a second-order model, we have to calculate where the maximum point is according to the fitted model. That is, finding the stationary point of the response surface. We have to notice that the fitted model is obtained by estimating the coefficient of equation (26), so the stationary point is a result of a fitted model, not the true response surface. If the stationary point locates in the original design space, we will perform Yield Analysis to obtain the yield of the stationary point. If not, we will discard the point. The principle to solve the stationary point according to the fitted second-order model is

$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial x_2} = \ldots = \frac{\partial y}{\partial x_k} = 0 \tag{27}$$

For example, now we have a fitted second-order model:

$$y = 36 + 3x_1 + 5x_2 - 2x_1^{\;2} - 4x_2^{\;2} - 8x_1 x_2$$

we have

$$\frac{\partial y}{\partial x_1} = 3 - 4x_1 - 8x_2 = 0 \text{, and } \frac{\partial y}{\partial x_2} = 5 - 8x_2 - 8x_1 = 0$$

so we obtain the stationary point of the fitted second-order model

$$x_1 = \frac{1}{2},\quad x_2 = \frac{1}{8}$$

3.3.6 Correlation Terms Handling

In analog circuit, there exist certain structures that transistors in the structure are correlated with each other. For example, we have to match the widths of transistors in a current mirror. That is, the widths of transistors in a current mirror should be positive correlated (one increases (decreases), the other increases (decreases)). These transistors are considered as correlation terms in our optimization tool. We treat the transistors that are highly correlated as one group to prevent possible mismatch. Users can assign the group member and indicate the relationships between transistors in the group are positive-correlated or negative correlated. If two transistors are assigned to be positive correlated, once width of a transistor is increased (decreased), width of the other transistor will be increased (decreased) meanwhile. If two transistors are assigned to be negative correlated, once width of a transistor is increased (decreased), width of the other transistor will be decreased (increased) at the same time. By this feature, our tool concerns the interaction of transistors in the circuit and therefore keeps the interaction information in the optimization process.

Chapter 4 Experimental Results

4.1 Circuits under Test

Four OLED drivers are used in this thesis to demonstrate the FlexiOptimizer. Figure 4.1 to Figure 4.4 shows the schematics of the OLED drivers circuit 1 to circuit 4, respectively. V_{data} in all circuits is the data input and V_{addr} is the selection signal. When the V_{data} and the V_{addr} are high, the TFT is turned on to drive OLED.

Circuit 1 is a two-TFT driver [He 01]. Transistor *M1* drives the gate input of *M2*. *M1* is the TFT to drive the OLED. Figure 4.2 shows a three-TFT driver [Lin 11]. Compared with circuit 1, circuit 2 has one additional transistor M3 to reduce the difference between voltage n1 and n2 caused by the leakage current. Circuit 3 is a four-TFT driver [Chen 09]. Circuit 4 is a five-TFT driver [Servati 02a]. To model the OLED, we choose the point that OLED current is about 10 μA according to the Figure 4.5, which is proposed in [Chen 09]. According to Figure 4.5, while the OLED current is 10μA, the V_{OLED} is about 8V. Thus, we model the OLED by the equivalent resistance as 800K Ohm. The spice model used in this thesis is a level 1 model. The parameters used in the model is shown in Table 4.1.

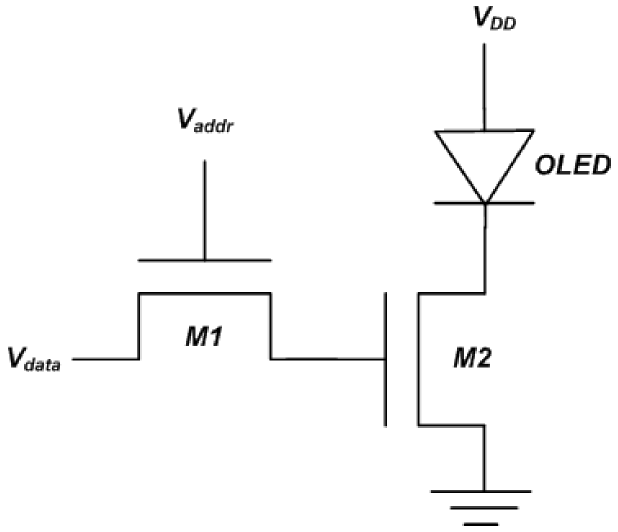


Figure 4.1 Schematic of Circuit 1

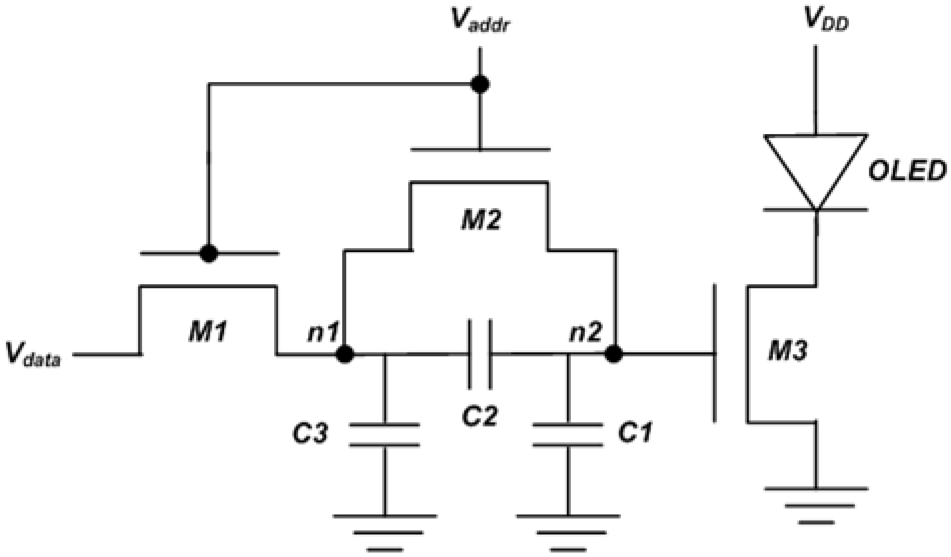


Figure 4.2 Schematic of Circuit 2

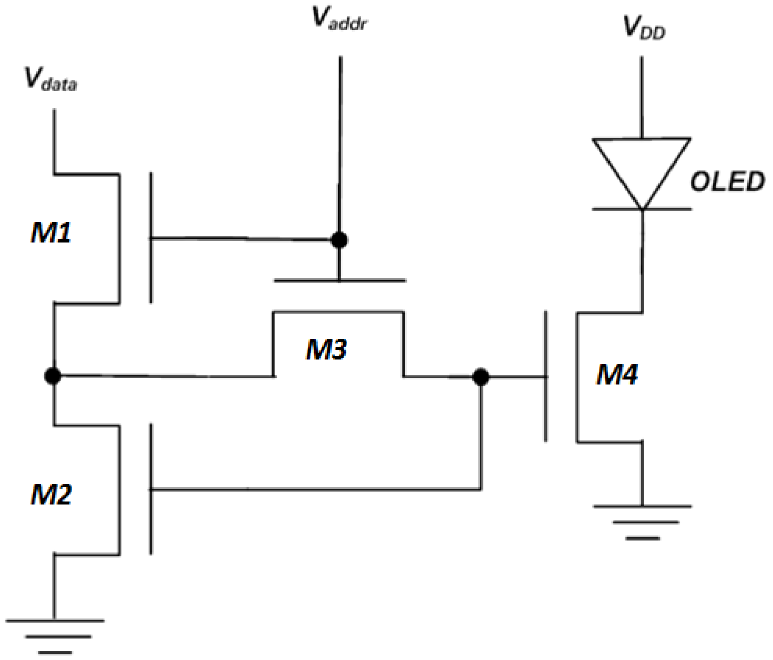


Figure 4.3 Schematic of Circuit 3

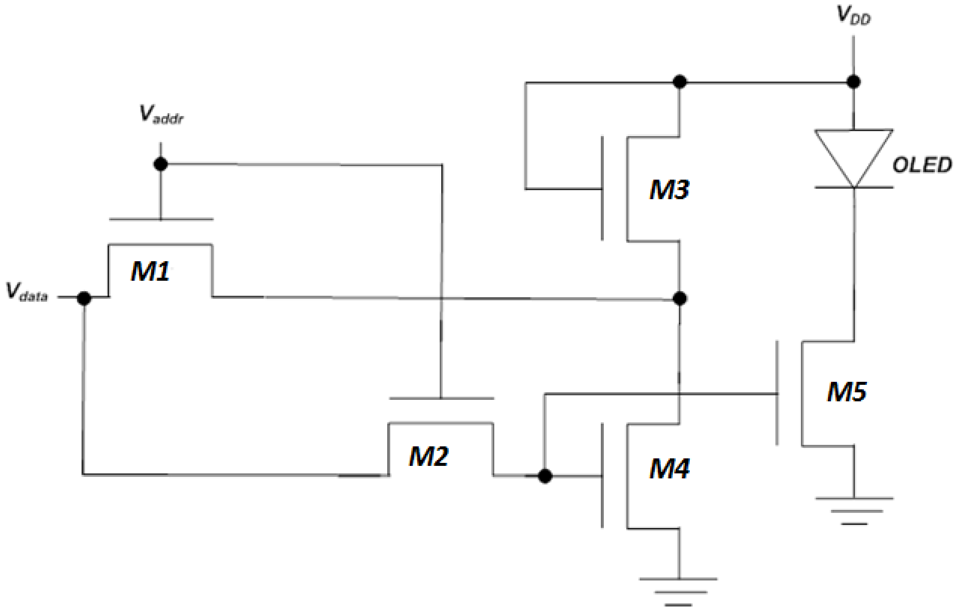


Figure 4.4 Schematic of Circuit 4

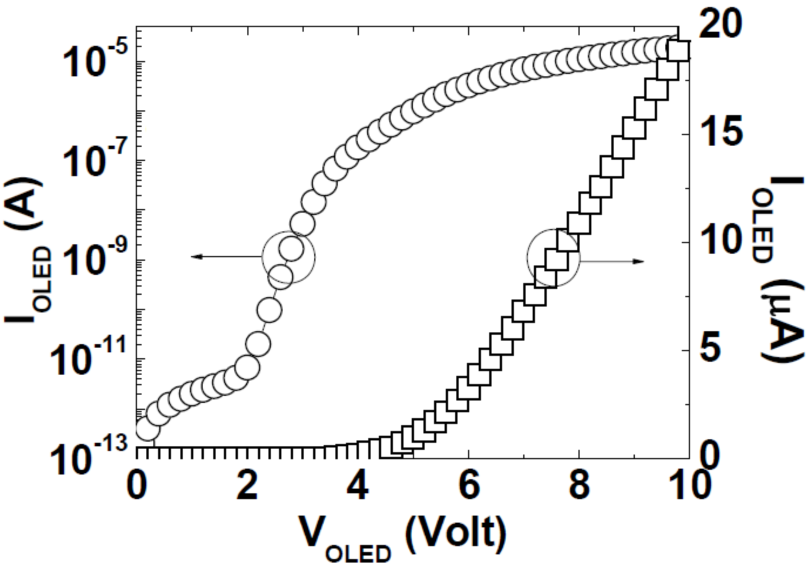


Figure 4.5 I-V curve of OLED [Chen 09]

Table 4.1 SPICE parameters used in level 1 model

Parameters	TOX	VTO	KN
a-Si	300nm	2V	7.5e-9
IGZO	80nm	1.24V	5.6e-7

4.2 Yield Optimization Results

4.2.1 OLED Drivers

We demonstrate our yield optimization results in both a-Si technology and IGZO technology. The results are compared with the circuits designed by the designer, making the specs to locate in the middle. The optimization settings for a-Si drivers are shown in Table 4.2. The constraints are set for width of each transistor and total width. The specification is I_{OLED} should be located within a region that the current is enough to light the OLED and is not so high to break the OLED. The parameters include the supply voltage, Monte Carlo trials, operation time (t_{total}), mobility range, inter-die variation, and intra-die variation. Since the lifetime for a-Si drivers is longer, the total operation time is set to be 1,728,000 seconds, that is, 20 days.

Table 4.2 Optimization settings for a-Si drivers

Constraints	10μm < W_i < 500μm W_{total} < 500μm
Specification	1μA < I_{OLED} < 20μA
Parameters	V_{DD}=25V, V_{data}=V_{addr}=15V *Monte Carlo trials* = 1,000 *Operation time* = 1,728,000 seconds -26% < *change in mobility* < 8% 1.8V < *Inter-die* V_{th} *variation* < 2.2V *σ of Intra-die* V_{th} *variation* = 0.5V

With the settings, we obtain the optimization result in Table 4.3. The table is divided into two parts. One is the yield results for designed circuits. The other is the yield results for optimized circuits. Each table includes the width of each transistor and the yield considering process variation, bending effect, and aging effect. The designed circuit is designed to locate I_{OLED} in the middle of the specification, that is, 10.5μA. We can see the optimized yields are 6.8% higher than the designed yields, in average.

Table 4.3 Optimization result for a-Si drivers

	$Yield_{Designed}$	W_{M1}	W_{M2}	W_{M3}	W_{M4}	W_{M5}
Circuit 1	61.2%	10μm	195μm	-	-	-
Circuit 2	57.1%	10μm	10μm	195μm	-	-

Circuit 3	82.1%	50μm	50μm	10μm	390μm	-
Circuit 4	58.0%	10μm	10μm	10μm	10μm	195μm
Average	64.6%	-	-	-	-	-

	$Yield_{optimized}$	W_{M1}	W_{M2}	W_{M3}	W_{M4}	W_{M5}
Circuit 1	65.8%	160μm	286μm	-	-	-
Circuit 2	64.0%	10μm	23μm	322μm	-	-
Circuit 3	90.0%	87μm	102μm	10μm	290μm	-
Circuit 4	65.7%	81μm	78μm	46μm	10μm	283μm
Average	71.4%	-	-	-	-	-

The optimization settings for IGZO drivers are shown in Table 4.4. The width of each transistor is smaller because the mobility of IGZO TFT is much larger than a-Si TFT, so it needs smaller W/L ratio. Since the width of each transistor is smaller, we set the maximum value of total width to be 300μm. Owing to shorter lifetime in IGZO technology, we set operation time to be 10,000 seconds to demonstrate.

Table 4.4 Optimization settings for IGZO drivers

Constraints	10μm < W_i < 100μm W_{total} < 300μm
Specification	1μA < I_{OLED} < 20μA
Parameters	V_{DD}=20V, V_{data}=V_{addr}=12V *Monte Carlo trials* = 1,000 *Operation time* = 10,000 seconds -2% < *change in mobility* < 3% 0.94V < *Inter-die* V_{th} *variation* < 1.54V *σ of Intra-die* V_{th} *variation* = 0.5V

With the settings, we obtain the optimization result in Table 4.5. Table 4.5 is divided into two parts. One is the yield results for designed circuits. The other is the yield results for optimized circuits. Each table includes the width of each transistor and the yield considering process variation, bending effect, and aging effect. The designed circuit is designed to locate I_{OLED} in the middle of the specification, that is, 10.5μA. We can see the optimized yield is average 12.0% higher than designed yield. The number is larger than that of a-Si drivers. The reason is the IGZO drivers are more sensitive to aging effect. The harder the effects can be considered when designing the circuits, the better performance FlexiOptimizer will have. According to the results, we can deduce that considering the three effects, the best design may not be located in the middle of the specs anymore.

Table 4.5 Optimization result for IGZO drivers

	$Yield_{Designed}$	W_{M1}	W_{M2}	W_{M3}	W_{M4}	W_{M5}
Circuit 1	75.3%	10μm	10μm	X	X	X
Circuit 2	69.0%	10μm	10μm	10μm	X	X
Circuit 3	75.3%	10μm	10μm	10μm	30μm	X
Circuit 4	81.2%	10μm	10μm	10μm	10μm	10μm
Average	75.2%	-	-	-	-	-

	$Yield_{optimized}$	W_{M1}	W_{M2}	W_{M3}	W_{M4}	W_{M5}
Circuit 1	85.6%	70µm	12µm	X	X	X
Circuit 2	85.7%	35µm	11µm	13µm	X	X
Circuit 3	87.5%	99µm	93µm	10µm	16µm	X
Circuit 4	90.0%	65µm	88µm	42µm	48µm	12µm
Average	87.2%	-	-	-	-	-

In these experiments, the total runtime for all drivers are less than 150 seconds. Most of the runtime is spent on the aging effect calculation, since the t_{total} is divided into several time intervals and 1,000 (= Monte Carlo trials) SPICE simulations are performed at the beginning of each time interval. Therefore, more than 10,000 SPICE simulations have to be performed in each yield analysis.

4.2.2 Differential Operational Amplifier

In this subsection, we demonstrate the tool by optimizing differential operational amplifier. The schematic of the circuit is shown in Figure 4.6. We can observe that some transistors should be paired together to avoid the mismatch. In our tool, we support the feature to pair transistors together. In this case, we pair (M_1,M_2), (M_3,M_4), (M_6,M_7), (M_8,M_9), (M_{11},M_{12}), and (M_{13},M_{14}), respectively. We call them *correlation pairs*. In a correlation pair, the width of each transistor will be set to the same value. Table 4.6 shows the optimization settings of the circuit in a-Si technology. Table 4.7

shows the optimization settings of the circuit in IGZO technology. Table 4.8 and Table 4.10 show the optimization results of the circuit in a-Si and IGZO technology, respectively. In a-Si technology, the yield is improved from 27% to 35%. The total number of RSM points used to optimize this circuit is 32, the corresponding runtime is 5182 seconds. In IGZO technology, the yield is improved from 22% to 57%. The total number of RSM points used to optimize this circuit is 32, the corresponding runtime is 4789 seconds. In the last row of Table 4.10, we show the result without setting the correlation pairs. We can observe that the yield is low since differential operational amplifier requires the schematic to be symmetric. Thus, correlation terms handling is an important feature in FlexiOptimizer.

The widths of designed circuit and optimized circuit for a-Si and IGZO technology are shown in Table 4.9 and Table 4.11, respectively. The widths of the circuit optimized without setting correlation pairs are shown in Table 4.12. From Table 4.9 and Table 4.11, we can see our tool can handle the correlation pairs correctly. The widths of the transistors in each pair are all the same. Since there are six pairs, the number of input variables is reduced to 14 – 6 = 8. After screening experiment, the number of input variables is further reduced to 7, and 5 in a-Si and IGZO, respectively. From Table 4.12, we can observe that the circuit optimized without setting correlation pairs is almost symmetric, but it still has low yield. We also can see that the number of important variables in this case is 7.

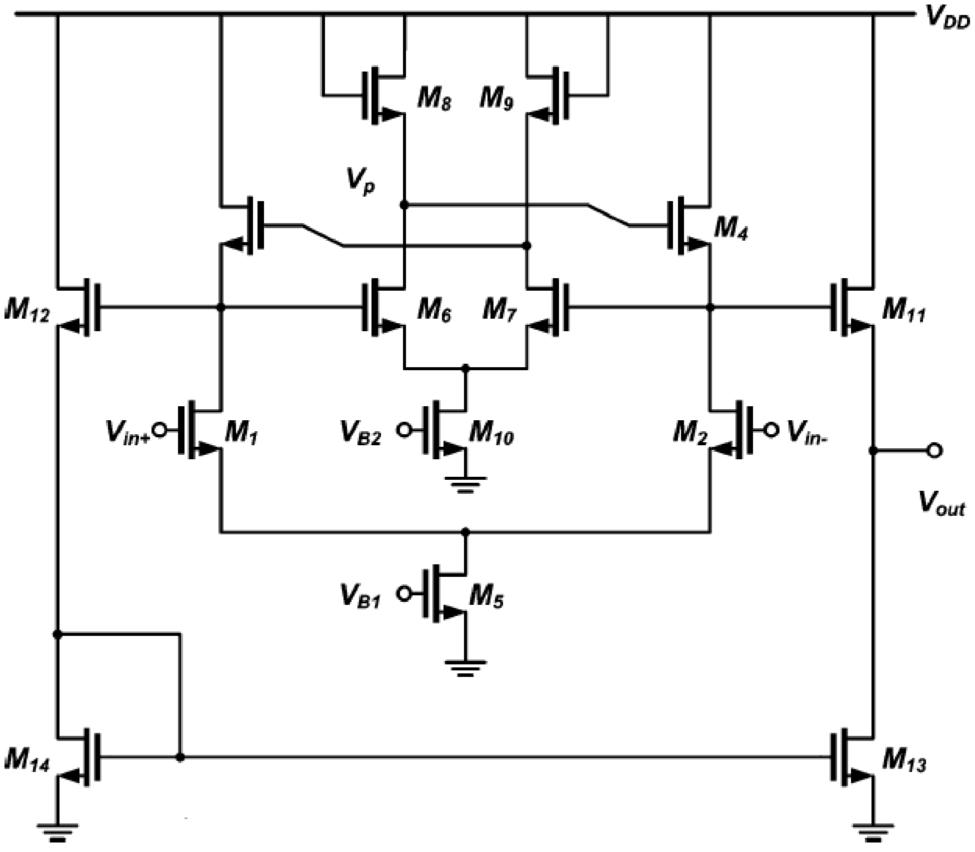


Figure 4.6 Schematic of differential operational amplifier [Tarn 10]

Table 4.6 Optimization settings for a-Si OPAMP

Constraints	10μm < W_i < 500μm
Specifications	*Gain* > 10 *Band width* > 5,000 Hz *Phase margin* > 55^o
Parameters	V_{DD}=25V, V_{in1}=V_{in2}=11.5V, V_{b1}=3.1V, V_{b2}=2.5V *Monte Carlo trials* = 100 *Operation time* = 1,000 seconds -26% < *change in mobility* < 8% 1.8V < *Inter-die* V_{th} *variation* < 2.2V *σ of Intra-die* V_{th} *variation* = 0.5V

Table 4.7 Optimization settings for IGZO OPAMP

Constraints	10µm < W_i < 100µm
Specifications	*Gain* > 10 *Band width* > 5,000 Hz *Phase margin* > 55^{o}
Parameters	$V_{DD}=25V$, $V_{in1}=V_{in2}=11.5V$, $V_{b1}=3.1V$, $V_{b2}=2.5V$ *Monte Carlo trials* = 100 *Operation time* = 1,000 seconds -2% < *change in mobility* < 3% 0.94V < *Inter-die* V_{th} *variation* < 1.54V *σ of Intra-die* V_{th} *variation* = 0.5V

Table 4.8 Optimization results for a-Si OPAMP

a-Si	Yield%	RSM points used	Runtime (sec)
Designed	27%	-	-
Optimized	35%	32	5,182

Table 4.9 Widths for designed and optimized a-Si OPAMP

a-Si	$W_{M1,M2}$	$W_{M3,M4}$	W_{M5}	$W_{M6,M7}$	$W_{M8,M9}$	W_{M10}	$W_{M11,M12}$	$W_{M13,M14}$
Designed	125	94	125	250	125	250	500	63
Optimized	328	40	487	203	221	387	10	22

unit of W_{Mi} : µm

Table 4.10 Optimization results for IGZO OPAMP

IGZO	Yield%	RSM points used	Runtime (sec)
Designed	22%	-	-
Optimized	57%	32	4,789
Optimized (no correlation)	7%	42	4,541

Table 4.11 Widths for designed and optimized IGZO OPAMP

IGZO	$W_{M1,M2}$	$W_{M3,M4}$	W_{M5}	$W_{M6,M7}$	$W_{M8,M9}$	W_{M10}	$W_{M11,M12}$	$W_{M13,M14}$
Designed	25	19	25	50	25	50	100	13
Optimized	10	37	10	83	75	10	18	90

unit of W_{Mi} : μm

Table 4.12 Widths for optimized IGZO OPAMP without setting correlation pairs

W_{M1}	W_{M2}	W_{M3}	W_{M4}	W_{M5}	W_{M6}	W_{M7}
71	70	61	10	10	10	10
W_{M8}	W_{M9}	W_{M10}	W_{M11}	W_{M12}	W_{M13}	W_{M14}
10	10	65	10	12	45	46

unit of W_{Mi} : μm

4.2.3 Optimize New/Bent or Aged/Bent

We wonder how different between the yields if we optimize the yield of new/bent circuit and aged/bent circuit. Thus, we use circuit 4 in a-Si technology and IGZO technology to demonstrate the thought. The optimization settings are the same as Table 4.2 and Table 4.4 for a-Si and IGZO, respectively. The results are shown in Table 4.12 and Table 4.13.

Table 4.12 Yield of optimizing new/bent circuit and aged/bent circuit (a-Si)

	Optimize New/Bent	Optimize Aged/Bent
$\text{Yield}_{\text{NEW}}$	100.0%	99.9%
$\text{Yield}_{\text{AGED}}$	40.1%	65.7%

Table 4.13 Yield of optimizing new/bent circuit and aged/bent circuit (IGZO)

	Optimize New/Bent	Optimize Aged/Bent
$\text{Yield}_{\text{NEW}}$	99.9%	97.1%
$\text{Yield}_{\text{AGED}}$	81.1%	90.0%

From Table 4.12 and Table 4.13 we can observe that the yield may be a bit lower when the circuit is new if we optimize the aged/bent circuit. But after the operation, it has higher yield than new/bent optimized circuit. That is, if we optimize the yield of the circuit in aged/bent condition, the lifetime of the circuit will be longer than the circuit we optimize the yield in new/bent condition.

4.3 Technique Effectiveness Analysis

In this section, we analyze the effectiveness of the proposed technique. We use circuit 4 in a-Si technology to demonstrate. The optimization settings are the same as Table 4.2. In the first experiment, we want to see if screening experiment is effective. We perform 10 times of optimization under the aged/bent condition with and without using screening experiment, respectively. We observe the average of the yields, the standard deviation of the yields and the average of RSM points used. The result is shown in Table 4.14.

Table 4.14 Yield optimization w/ and w/o screening experiment in 10 times

	Mean Yield	σ of yield	Worst Yield	Best Yield	RSM points used
w/o screening experiment	62.3%	4.11%	51.6%	67.0%	21.5
w/ screening experiment	66.2%	1.26%	64.1%	68.3%	35.7

From the experiment, we observe that using the screening experiment, the yield is average 3.9% higher than without using the screening experiment. The σ of the third column indicates the standard deviation. With using the screening experiment, the standard deviation of optimized yield is only 1.26%. Without the screening experiment, the standard deviation of optimized yield is 4.11%. The worst optimized yield without screening experiment is 51.6%, a relatively low yield. We can deduce that with using screening experiment, the optimized results will be more stable. Although 14.2 more RSM points are used with screening experiment, the more stable and better result provided by using screening experiment is still attractive. We believe screening experiment may reduce the RSM points used when the number of input variables is high.

In the second experiment, we want to know how many new RSM points are needed to fit first-order model so that we can reach the higher yield effectively. We use circuit 4 in a-Si technology to demonstrate. The optimization settings are the same as Table 4.2. We perform the experiment w/ and w/o using screening experiment. Therefore, we can observe if screening experiment will affect the RSM points needed to fit first-order model. The result is shown in Figure 4.7. and Figure 4.8. We perform five times of optimization and see the average of the yield and the maximum yield reached in these five times of optimization. The x-axis is the number of new RSM points generated before model-fitting. The y-axis is yield.

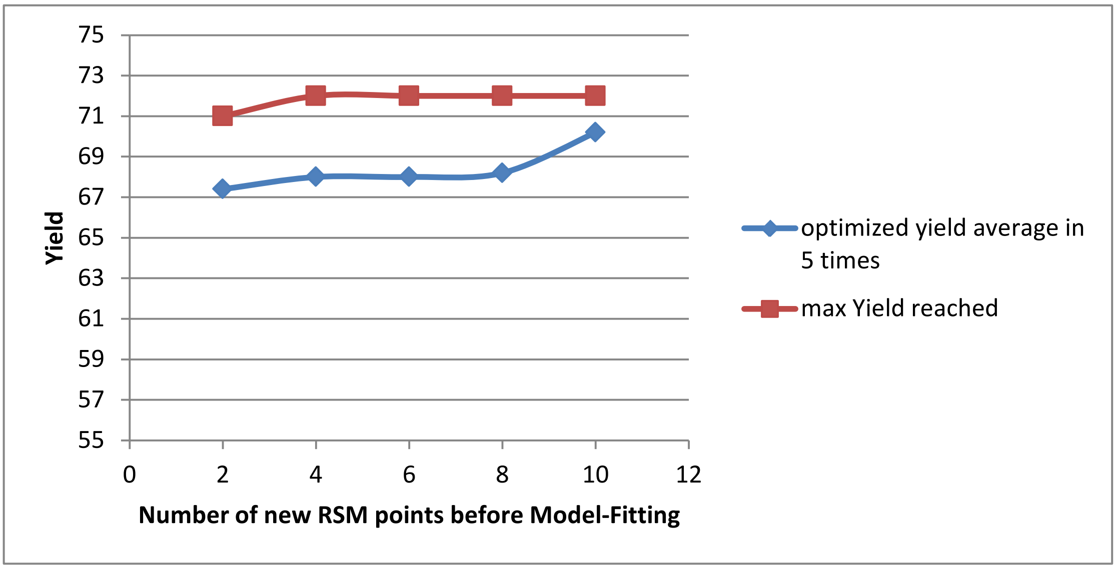


Figure 4.7 Yield to number of new RSM points w/ screening experiment

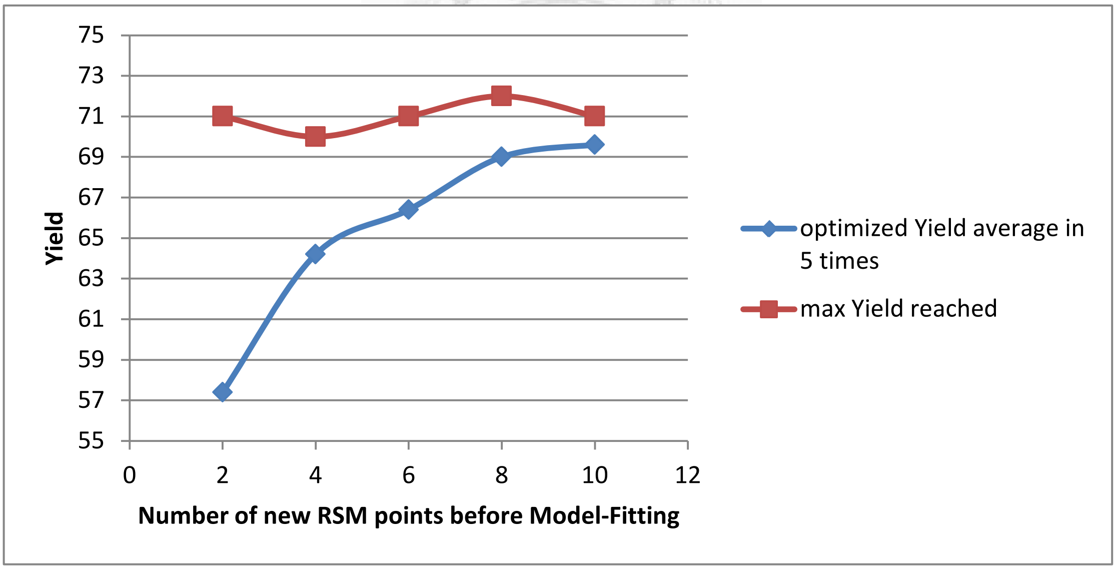


Figure 4.8 Yield to number of new RSM points w/o screening experiment

In this experiment, we can observe that if we use screening experiment, the average yield and the maximum yield reached are less relative to the number of new

RSM points. That is, the optimization result is stable if we applied the screening experiment. If we do not apply the screening experiment, we can observe that the average yield is relatively low if we use less number of new RSM points. That is, the optimization result are more relative to the number of new RSM points if we do not use the screening experiment. We can also observe that the maximum yield reached is not so relative to the number of new RSM points. That is because the RSM flow still keep the possibility to find the optimum region although less RSM points used.

Chapter 5 Conclusion and Future Work

A yield optimization tool, FlexiOptimizer, for flexible TFT analog circuits is proposed in this thesis. Three important effects for flexible TFT circuits are considered. They are 1) process variation of V_{th0}, 2) Aging effect, and 3) bending effect. The process variation is considered as inter-die variation and intra-die variation in FlexiOptimizer. The aging effect is modeled as ΔV_{th}. Two aging models for a-Si and IGZO technology are used. Bending effect is modeled as mobility change. Circuit should pass the specs under both conditions: 1) maximum possible mobility and 2) minimum possible mobility. The optimization algorithm used in this thesis is response surface methodology. An orthogonal-array-based screening experiment to identify important variables is proposed. FlexiOptimizer can also handle the correlation terms if users want to have some input variables with the same value during the optimization flow. Two designs (OLED drivers and differential operational amplifier) in two technologies (8μm a-Si and 10μm IGZO technology) are demonstrated in the experimental results. According to the experimental results, the optimized yield is 12.0% and 6.8% higher than designed yield of IGZO and a-Si OLED drivers in average, respectively.

The aging effect calculation in the proposed tool takes much time. Most of the runtime of FlexiOptimizer is spent on the aging effect calculation. Thus, a speedup mechanism can be added to FlexiOptimizer. One possible way to speed up the aging effect calculation is using a statistical technique to predict the trend of the ΔV_{th}, such as time series analysis. Another possible way is to speed up the simulator. Since the SPICE used in FlexiOptimizer is developed by our own, we can apply techniques directly to SPICE. Since the parameters of the circuits we have to simulate are similar, the conductance matrices used to solve the circuit may also be similar. A clustering method may be applied to cluster the points with similar conductance matrices so that we can reduce the matrix calculation efforts.

Reference

[Aggarwal 07] V. Aggarwal, “Analog Circuit Optimization using Evolutionary Algorithms and Convex Optimization, ” *MS. thesis*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2007.

[Alvarez 88] A. R. Alvarez, “Application of Statistical Design and Response Surface Methods to Computer-Aided VLSI Device Design, ” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, Issue. 2, pp.272-288, Feb. 1988.

[Aoki 96] H. Aoki, “Dynamic Characterization of a-Si TFT-LCD pixels, ” *IEEE Transactions on Electron Devices*, vol.43, no.1, Jan. 1996.

[Bae 06] B. S. Bae, J. W. Choi, J. H. Oh, and J. Jang, “Level shifter embedded in drive circuits with amorphous silicon TFTs,” *IEEE Transactions on Electron Devices*, vol.53, no.3, pp. 494-498, March. 2006.

[Bai 08] J. Y. Bai, H. R. Chen, and S. Y. Chien, “An asynchronous fixed-coefficient FIR filter implemented with flexible a-Si TFT technology,” *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, pp.1790-1793, Nov. 2008.

[Box 51] G. E. P. Box, and K. B. Wilson, “On the Experimental Attainment of Optimum Conditions, ” *Journal of the Royal Statistical Society. Series B*, vol. 13, No. 1, pp.1-45, 1951.

[Boyd 07] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, Arash Hassibi, “A tutorial on geometric programming, ” *Optimization and Engineering, Stanford University*, 2007.

[Carley 04] K. M. Carley, N. Y. Kamneva, and J. Reminga, “Response Surface Methodology, ” *Technical Report, CASOS, Carnegie Mellon University*, CMU-ISRI-04-136, Oct. 2004.

[Chen 09] C. Chen, J. Kanicki, K. Abe, and H. Kumomi, “AM-OLED Pixel Circuits

Based on a-InGaZnO Thin Film Transistors, ” *Journal of the Society for Information Display*, vol. 17, Issue 6, pp. 525-534, Jun. 2009.

[Cheng 12] Y. F. Cheng, L. Y. Chan, Y. L. Chen, Y. C. Liao, and C. N. J. Liu, “A Bias-Driven Approach to Improve the Efficiency of Automatic Design Optimization for CMOS Op-Amps,” *IEEE Asia Symposium on Quality Electronic Design,* July. 2012.

[Cheon 08] J. H. Cheon, J. H. Bae, and J. Jann, “Mechanical stability of poly-Si TFT on metal foil,” *Solid-State Electronics*, vol. 52, issue 3, pp. 473-477, March. 2008.

[Cho 11] E. N. Cho, C. E. Kim, and I. Yun, “Analysis of Bias Stress Instability in Amorphous InGaZnO Thin-Film Transistors, ” *IEEE Transactions on Device and Materials Reliability*, vol.11, no.1, Mar. 2011.

[Dehuff 05] N.L. Dehuff, E.S. Kettenring, D. Hong, H.Q. Chiang, J.F. Wager, R.L. Hoffman, C.H. Park, and D.A. Keszler, “Transparent thin-film transistors with zinc indium oxide channel layer,” *Journal of Applied Physics*, vol.97, Mar. 2005.

[Duffin 62] R. J. Duffin, and E. L. Peterson, “Constrained Minima Treated by Geometric Means, ” *Westinghouse Scientific Paper*, 64-158-129-P3, 1964.

[Duffin 66] R. J. Duffin, and E. L. Peterson, “Duality Theory for Geometric Programming, ” *SIAM Journal on Applied Mathematics*, vol. 14, No. 6, pp. 1307-1349, Nov. 1966.

[Fortunato 04] E.M.C. Fortunato, P.M.C. Barquinha, A.C.M.B.G. Pimentel, A.M.F. Goncalves, A.J.S. Marques, R.F.P. Marins, and L.M.N. Pereira, “Wide-bandgap high-mobility ZnO thin-film transistors produced at room temperature, ” *Applied Physics Letters,* pp.254, 2004.

[Fujii 01] N. Fujii and H. Shibatu, “Analog Circuit Synthesis by Superimposing of Sub-Circuits, ” *Proceedings of the International Symposium on Circuits and Systems*, vol. 5, No. 1, pp. 427-430, 2001.

[Fung 10] T. C. Fung, “Amorphous In-Ga-Zn-O Thin Film Transistor for Future Optoelectronics,” *Ph.D. thesis*, Dept. Elect. Eng., University of Michigan, 2010.

[Gielen 90] G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing," *IEEE Journal of Solid-State Circuits*, vol.25, no.3, pp.707-713, Jun. 1990.

[Gleskova 99] H. Gleskova, S. Wagner, and Z. Suo, "Failure resistance of amorphous silicon transistors under extreme in-plane strain," *Applied Physics Letters*, vol.75, no.19, pp.3011-3013, Nov. 1999.

[Gleskova 02] H. Gleskova, S. Wagner, W. Soboyejo, and Z. Suo, "Electrical response of amorphous silicon thin-film transistors under mechanical strain," *Journal of Applied Physics*, vol.92, no.10, pp.6224-6229, Nov. 2002.

[Graeb 01] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich, "The Sizing Rules Method for Analog Integrated Circuit Design, " *International Conference on Computer-Aided Design*, Proceedings, pp.343-349, 2001.

[Grimbleby 00] G. B. Grimbleby, "Automated Analogue Circuit Synthesis using Genetic Algorithms," *IEEE Proceedings of Circuits, Devices and Systems*, vol. 147, Issue 6, pp.319-323, Dec. 2000.

[Hack 89] M. Hack, M. S. Shur, and J. G. Shaw, "Physical models for amorphous-silicon thin-film transistors and their implementation in a circuit simulation program," *Electron Devices, IEEE Transactions on*, vol.36, no.12, pp.2764-2769, Dec. 1989.

[He 01] Y. He, R. Hattori, and J. Kanicki, "Improved A-Si:H TFT Pixel Electrode Circuits for Active-Matrix Organic Light Emitting Display, " *IEEE Transactions on Electron Devices*, vol. 48, no. 7, July. 2001.

[Hedayat 99] A. S. Hedayat, N. J. A. Sloane, J. Stufken, "Orthogonal Arrays – Theory and Applications, " Springer, 1999.

[Hershenson 99] Maria del Mar Hershenson, Sunderarajan S. Mohan, Stephen P. Boyd, Thomas H. Lee, "Optimization of Inductor Circuits via Geometric Programming, " *Design Automation Conference (DAC), 36th ACM/IEEE,* 1999.

[Hill 66] W. Hill and W. Hunter, "A Review of Response Surface Methodology: A

Literature Survey," *Technometrics*, Vol. 8, No. 4, pp. 571-590, 1966.

[Huang 10] T. C. Huang, Kenjiro Fukuda, C. M. Lo, Y. H. Yeh, Tsuyoshi Sekitani, Takao Someya, and K. T. Cheng, "Pseudo-CMOS: A Novel Design Style for Flexible Electronics", *Design, Automation & Test in Europe*, 2010.

[Ishida 10] K. Ishida, N, Masunaga, R. Takahashi, T. Sekitani, S. Shino, U. Zschieschang, H. Klauk, M. Takamiya, T. Someya, and T. Sakurai, T, "User Customizable Logic Paper (UCLP) with organic sea-of-transmission-gates (SOTG) architecture and ink-jet printed interconnects," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pp.138-139, 7-11 Feb. 2010.

[Jain 05] K. Jain, M. Klosner, M. Zemel, and S. Raghunandan, "Flexible Electronics and Displays: High-Resolution, Roll-to-Roll, Projection Lithography and Photoablation Processing Technologies for High-Throughput Production," *Proceedings of the IEEE*, vol.93, no.8, pp.1500-1510, Aug. 2005.

[Jang 99] Jang et al. United States Patent, number 5923050, Jul. 13, 1999.

[Kim 06] C. H. Kim, S. J. Yoo, H. J. Kim, J. M. Jun, and J. Y. Lee, "Integrated a-Si TFT row driver circuits for high-resolution applications," *Journal of Society for Information Display*, vol. 14, no. 4, pp.333-337, May. 2006.

[Kim 09] G. H. Kim, H. S. Kim, H. S. Shin, B.D. Ahn, K.H. Kim, and H.J. Kim, "Inkjet-Printed InGaZnO Thin Film Transistor, " *Journal of Thin Solid Films*, vol. 517, pp. 4007-4010, May. 2009.

[Kim 12] S. J. Kim, S. Y. Lee, Y. W. Lee, S. H. Kuk, J. Y. Kwon, and M. K. Han, "Effect of Charge Trapping/Detrapping on Threshold Voltage Shift of IGZO TFTs under AC Bias Stress," *Electrochemical and Solid-State Letters*, vol. 15, pp. H108-H110, Feb. 2012.

[Kirkpatrick 83] S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 13, May, 1983.

[Kumar 04] A. Kumar, S. Sambandan, K. Sakariya, P. Servati, and A. Nathan, "Amorphous silicon shift registers for display drivers," *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol.22, no.3, pp.981-986, May. 2004.

[Kumar 05] A. Kumar, A. Nathan, and G. E. Jabbour, "Does TFT mobility impact pixel size in AMOLED backplanes?," *Electron Devices, IEEE Transactions on*, vol.52, no.11, pp. 2386- 2394, Nov. 2005.

[Leung 01] Yiu-Wing Leung,"An Orthognal Genetic Alrotithm with Quantization for Global Numerical Optimization" *IEEE Transactions on Evolutionary Computation*, vol. 5, No.1, Feb. 2001.

[Libsch 93] F. R. Libsch and J. Kanicki, "Bias-stress-induced stretched-exponential time dependence of charge injection and trapping in amorphous thin-film transistors," *Appl. Phys. Lett.*, vol. 62, no. 11, pp. 1286–1288, Mar. 1993.

[Lin 11] Chen-Wei Lin, Mango C.-T. Chao, and Yen-Shih Huang, "A Novel Pixel Design for AM-OLED Displays Using Nanocrystalline Silicon TFTs", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 19, No. 6, June, 2011.

[Liu 09] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, F. Fernandez, "Analog circuit optimization system based on hybrid evolutionary algorithms, " *VLSI Journal of Integration*, vol. 42, issue 2, pp. 137-148, Feb. 2009.

[Ma 11] E. H. Ma, "DC Analysis for Flexible TFT Circuits Considering Process Variation, Aging, and Bending Effects," M.S. thesis, Dept. Graduate Institute of Elect. Eng., National Taiwan Univ. Sep. 2010.

[Madeira 97] P. Madeira and R. Hornsey, "Analog circuit design using amorphous silicon thin film transistors," *Electrical and Computer Engineering, 1997. IEEE 1997 Canadian Conference on*, vol.2, pp.633-636, May. 1997.

[Mohan 02] N. Mohan, K. S. Karim, and A. Nathan, "Design of multiplexer in amorphous silicon technology," *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol.20, no.3, pp.1043-1047, May. 2002.

[Mohanty 12] S. P. Mohanty, J. Singh, E. Kougianos, and D. K. Pradhan, "Statistical

DOE-ILP Based Power-Performance-Process (P3) Optimization of Nano-CMOS SRAM, " *VLSI Journal of Integration,* vol. 45, Issue 1, pp. 33-45, Jan. 2012.

[Munzenrieder 11] Niko Munzenrieder, Kunigunde H. Cherenack, and Gerhard Troster, "The effects of mechanical bending and illumination on the performance of flexible IGZO TFTs," *IEE Transactions on Electron Devices,* vol. 58, No.7, July. 2011.

[Myers 91] R. H. Myers, "Response Surface Methodology in Quality Improvement," *Communications in Statistics – Theory and Methods,* vol.20, no.2, pp.457-476, 1991.

[Myers 02] R. H. Myers and D. Montgomery, "Response Surface Methodology: Process and Product Optimization Using Designed Experiments, " second edition, A Wiley-Interscience publication, 2002.

[Nomura 06] K. Nomura, H. Ohta, A. Takagi, T. Kamiya, M. Hirano, and H. Hosono, "Amorphous Oxide Semiconductors for High-Performance Flexible Thin-Film Transistors, " *Journal of Applied Physics,* vol. 45, pp.4303-4308, 2006.

[Nathan 04] A. Nathan, A. Kumar, K. Sakariya, P. Servati, S. Sambandan, and D. Striakhilev, "Amorphous Silicon Thin Film Transistor Circuit Integration for Organic LED Displays on Glass and Plastic," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 9, pp. 1477-1486, 2004.

[Noren 01] K. Noren and J. Ross, "Analog Circuit Design Using Genetic Algorithms, " *EE Times,* Aug. 2001.

[Park 09] J.-S. Park, T.-W. Kim, D. Stryakhilev, J.-S. Lee, S.-G. An, Y.-S. Pyo, D.-B. Lee, Y. G. Mo, D.-U. Jin, and H. K. Chung, "Flexible Full Color Organic Light-Emitting Diode Display on Polyimide Plastic Substrate Driven by Amorphous Indium-Gallium Zinc Oxide Thin-Film Transistors," *Appl.Phys. Lett.*, vol. 95, no. 1, pp. 013 503, Jul. 2009.

[Pengelly 02] J. Pengelly, "Monta Carlo Methods, " *University of Utago,* Feb. 2002.

[Powell 83] M. J. Powell, "Charge trapping instabilities in amorphous silicon–silicon nitride thin-film transistors,” *Appl. Phys. Lett.*, vol. 43, no. 6, pp. 597–599, Sep. 1983.

[Rao 46] C. R. Rao, “Hypercubes of Strength "d" Leading to Confounded Designs in Factorial Experiments,“ *Bulletin of the Calcutta Mathematical Society*, vol. 38 . pp. 67-78 1946.

[Rao 47] C. R. Rao, “Factorial Experiment Derivable from Combinatorial Arrangements of Arrays, ” *Supplement to the Journal of the Royal Statistical Society,* vol. 9, no. 1, pp. 128-139, 1947.

[Rao 49] C. R. Rao, “On a Class of Arrangements, ” *Proceedings of the Edinburgh Mathematical Society*, pp. 119-125, Dec. 1949.

[Servati 02a] P. Servati, S. Prakash, A. Nathan, and C. Py, “Amorphous silicon driver circuits for organic light-emitting diode displays,” *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, vol.20, no.4, pp.1374-1378, July. 2002.

[Servati 02b] P. Servati and A. Nathan, “Modeling of the reverse characteristics of a-Si:H TFTs,” *Electron Devices, IEEE Transactions on*, vol. 49, pp. 812-819, 2002.

[Servati 02c] P. Servati and A. Nathan, “Modeling of the static and dynamic behavior of hydrogenated amorphous silicon thin-film transistors,” *Journal of Vacuum Science and Technology A, Vacuum, Surfaces, and Films*, vol. 20, pp. 1038-1042, 2002.

[Sekitani 05a] T. Sekitani, S. Iba, Y. Kato, Y. Noguchi, T. Someya, T. Sakurai, “Ultraflexible organic field-effect transistors embedded at a neutral strain position,” *Applied Physics Letters*, vol.87, no.17, pp.173502-173502-3, Oct. 2005.

[Sekitani 05b] T. Sekitani, S. Iba, Y. Kato, Y. Noguchi, T. Someya, T. Sakurai, “Ultraflexible organic field-effect transistors embedded at a neutral strain position,” *Applied Physics Letters*, vol.87, no.17, pp.173502-173502-3, Oct. 2005.

[Shringarpure 07] Rahul Shringarpure, Sameer Venugopal, Zi Li, Lawrence T. Clark, David R. Allee, Edward Bawolek, and Daniel Toy, “Circuit Simulation of

Threshold-Voltage Degradation in a-Si: H TFTs Fabricated at 175℃”, *IEEE Transactions on Electron Devices*, Vol.54, No. 7, July, 2007.

[Smedt 03] B. D. Smedt, and G. C. E. Gielen, “WATSON: Design Space Boundary Exploration and Model Generation for Analog and RFIC Design,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol.22, no.2, pp. 213-224, Feb. 2003.

[Su 93] H. Su, C. Michael, and M. Ismail, “Yield Optimization of Analog MOS Integrated Circuits Including Transistor Mismatch,” *International Symposium on Circuits and Systems*, vol. 3, pp. 1801-1804, May. 1993.

[Su 10] L. Y. Su, H. Y. Lin, S. L. Wang, Y. H. Yeh, C. C. Cheng, L. H. Peng, and J. J. Huang, “Effects of Gate-Bias Stress on ZnO Thin-Film Transistors,” *Journal of the Society for Information Display*, vol. 18, pp. 802-806, Oct. 2010.

[Sung 10] S. Y. Sung, J. H. Choi, U. B. Han, K. C. Lee, J. H. Lee, J. J. Kim, W. Lim, S. J. Pearton, D. P. Norton, and Y. W. Heo, “Effect of Ambient Atmosphere on The Transfer Characteristics and Gate-bias Stress Stability of Amorphous Indium-Gallium-Zinc Oxide Thin-Film Transistors,” *Applied Physics Letters*, vol.96, pp.102107-1, 2010.

[Suresh 08] A. Suresh and J. F. Muth, “Bias Stress Stability of Indium Gallium Zinc Oxide Channel Based Transparent Thin-Film Transistors, ” *Applied Physics Letters*, vol. 92, pp. 033502-033502.3, Jan. 2008.

[Takagi 05] A. Takagi, K. Nomura, H. Ohta, H. Yanagi, T. Kamiya, M. Hirano, and H. Hosono, “Carrier transport and electronic structure in amorphous oxide semiconductor, a-InGaZnO4,” *Journal of Thin Solid Films*, vol. 486, pp.38-41, 2005.

[Tarn 10] Y. C. Tarn, P. C. Ku, H. H. Hsieh, and L. H. Lu, “An Amorphous-Silicon Operational Amplifier and Its Application to a 4-Bit Digital-to-Analog Converter,” *IEEE Journal of Solid-State Circuits*, vol.45, no.5, pp.1028-1035, May. 2010.

[Taherzadeh-Sani 03] M. Taherzadeh-Sani, R. Lotfi, H. Zare-Hoseini, and O. Shoaei, "Design Optimization of Analog Integrated Circuits using Simulation-Based Genetic Algorithm," *IEEE International Symposium on Signals, Circuits, and Systems*, vol. 1, pp.73-76, Jul. 2003.

[Tang 93] Boxin Tang, "Orthogonal Array-Based Latin Hypercubes" *Journal of American Statistical Association*, vol. 88, No.424, Dec. 1993.

[Tiwary 06] S. K. Tiwary, P. K. Tiwary, and R. A. Rutenbar, "Generation of Yield-Aware Pareto Surfaces for Hierarchical Circuit Design Space Exploration, " *Design Automation Conference*, pp.31-36, 2006.

[Venugopal 07] S. M. Venugopal and D. R. Allee, "Integrated a-Si:H Source Drivers for 4″ QVGA Electrophoretic Display on Flexible Stainless Steel Substrate," *Display Technology, Journal of*, vol.3, no.1, pp.57-63, March. 2007.

[Web Resource 1] https://controls.engin.umich.edu/wiki/images/3/3f/L4_Array.jpg

[Web Resource 2] https://controls.engin.umich.edu/wiki/index.php/File:Selector.jpg

[Web Resource 3] www.gnu.org/software/gsl, *GNU Operating System Official Website.*

[Zan 10] H. W. Zan, W.T. Chen, C. W. Chou, C. C. Tsai, C. N. Huang, and H. W. Hsueh, "Low Temperature Annealing with Solid-State Laser on UV Lamp Irradiation on Amorphous IGZO Thin-Film Transistors, " *Electrochemical and Solid-State Letters*, vol. 13, pp. H144-H146, Mar. 2010.

[Zhang 11] J. Zhang, X. F. Li, J. G. Lu, Z. Z. Ye, L. Gong, P. Wu, J. Huang, Y. Z. Zhang, L. X. Chen, and B. H. Zhao, "Performance and Stability of Amorphous InGaZnO Thin-Film Transistors with a Design Device Structure, " *Journal of Applied Physics*, vol. 110, Oct. 2011.

[Zener 61] C. Zener, "A Mathematical Aid in Optimizing Engineering Designs, " *Proceedings of the National Academy of Sciences of the United States of America*, vol. 47, No. 4, pp. 537-539, Apr. 1961.

[Zhou 06] L. Zhou, A. Wanga, S. C. Wu, J. Sun, S. Park, and T. N. Jackson, "All-organic active matrix flexible display," *Applied Physics Letters*, vol.88, no.8, pp.083502-083502-3, Feb. 2006.

Appendix – User manual

A.1 Getting Started with FlexiOptimizer

A.1.1 What is FlexiOptimizer?

FlexiOptimizer is an RSM-based yield-optimizing tool for flexible TFT analog circuits. The circuit simulator used in FlexiOptimizer is SPICE. The yield is calculated considering three effects: (1) process variation in threshold voltage, (2) aging effect results in threshold voltage shift, and (3) bending effect result in mobility change. Since it is hard for designer to design the circuits considering so many effects at the same time, FlexiOptimizer is a good solution to solve the problem.

A.1.2 Content of FlexiOptimizer

There are five packages in FlexiOptimizer. The five packages are listed below.

(1) util: The implementations of all command lines are in this package.

(2) SPICE: The circuit simulator used in FlexiOptimizer.

(3) YieldAnalyzer: A SPICE-based yield analyzer considering process variation, aging effect, and bending effect.

(4) RSM: The optimization algorithm is implemented in this package.

(5) FlexiOptimizer: In this package, there is a main function to deal with the command-line interface.

A.1.3 How to Use FlexiOptimizer

(1) Compilation:

Please type “make” to build the whole program. All packages will be compiled. To clear the binary file and object files, please type ‘make clean’.

(2) Execution

To execute the command-line interface, please execute the binary “<TOP_DIR>/bin/FlexiOpt” after compilation. After the command-line interface is called up, user can type the commands or run FlexiOptimizer by shell script. To run FlexiOptimizer by shell script, please type “source <path_of_shell_script>”.

A.2 Command-Line Commands

There are 22 commands supported in FlexiOptimizer. They are described in detail as following. The commands can be divided into two parts: required and optional. User should set all the required command before performing optimization. User is not required to set optional commands.

Required command:

(1) read_spice <spice_file_path>

This command reads the spice deck and initializes the SPICE. The input argument should be the path of spice deck.

(2) set_Vth_variation_range <lowerbound> <upperbound>

This command sets the lower bound and the upper bound of V_{th} inter-die variation. The input arguments are the lower bound and the upper bound, respectively. The type of input arguments should be **double**. For example, we want to set the lower bound and the upper bound of a-Si technology to be 1.8V and 2.2V, respectively. We have to type "set_Vth_variation_range 1.8 2.2".

(3) set_Vth_percentage <value1> <value2> ... <valueN>

This command sets the number of intervals of V_{th} inter-die variation and the percentage of each interval. For example, given the lower bound and the upper bound of V_{th} inter-die variation. We want to use 10 intervals with the percentages 2%, 4%, 10%, 16%, 18%, 18%, 16%, 10%, 4%, 2%. We have to type "set_Vth_percentage 2 4 10 16 18 18 16 10 4 2". One thing has to be noticed is no matter how many intervals are used, the total number of the percentage should be 100. All input arguments should be the type **double**.

(4) set_monte_carlo_number <Monte Carlo number>

This command set the number of Monte Carlo trials. For example, if we want to have 1000 circuit with different V_{th0} to perform yield analysis, we have to type "set_monte_carlo_number 1000". The input argument should be the type **unsigned**.

(5) set_intradie_Vth_sigma <sigma>

This command sets the intra-die V_{th} sigma. If we want to set the sigma to be 0.5V, we should type "set_intradie_Vth_sigma 0.5". The input argument should be the type **double**.

(6) set_operation_time <operation time>

This command set the total operation time. For example, if we want to set the total operation time to be 10000 seconds, we have to type "set_operation_time 10000". The input argument should be the type **double**.

(7) set_transistor_parameter_range <param_name> <lowerbound> <upperbound>

This command set the range of SPICE transistor parameter. For example, if we want to set the range of the mobility change from -26%~8%, we have to type "set_transistor_parameter_range KN 0.74 1.08". The first input argument, parameter name, should be the type **char***. The other input arguments should be the type **double**.

(8) set_specs <spec_name> <lowerbound> <upperbound>

This command sets the specifications. One specification is set at one call of this command. A specification is stored as a structure with specification name, lower bound, and upper bound. FlexiOptimizer will judge the circuits pass/fail according to all specifications set when calculating the yield.   The specification name should be the same with user's .measure target in SPICE deck. For example, if we want to set the specification I_{OLED} to be within 1μA to 20μA, we have to type "set_specs bias 0.000001   0.00002" since the current of OLED is named bias in our SPICE deck. The first input argument, specification name, should be the type **char***. The other input arguments should be the type **double**. In our OLED driver case, the total widths is the total constraint.

(9) set_optimization_individual_constraint   <lowerbound>   <upperbound>

This command set the lower bound and upper bound to each optimization variable. For example, if we want to set the width of each transistor to be within 10μm to 100μm, we have to type "set_optimization_individual_constraint   0.00001 0.0001". The input arguments should be the type **double**. In our case, transistor width is the individual constraint. The unit used in our case is meter.

(10) set_optimization_total_constraint   <constraint>

This command set the constraint to the summation of all optimization variables. For example, if we want to set the total width to be less than 500μm, we have to type "set_optimization_total_constraint 0.0005". The input argument should be the type **double**.

(11) set_number_of_levels <number of levels>

This command set the number of OA levels. For example, to set the number of OA levels to be 2, we should type "set_number_of_levels 2". The value of levels is determined as follows. If the number of levels is *n*, the value of level *k* (*k* < *n*) is $(upperbound - lowerbound)\times\frac{k}{n+1}+lowerbound$. That is, divide the range into $n+1$ equal portions by *n* points. Value of each point is the value of the corresponding level. For example, the upper bound is 100. The lower bound is 10. Number of levels is 2. Then $level1=(100-10)\times\frac{1}{3}+10=40$, $level2=(100-10)\times\frac{2}{3}+10=70$. The input argument should be type **unsigned**.

(12) set_technology <technology number>

This command sets the technology used. Technology number 1 represents a-Si, while 2 represents IGZO. If we want to use IGZO technology, we should type "set_technology 2". Then the IGZO SPICE model and aging model will be used. The input argument should be 1 or 2.

(13) optimize

This command is used after all required commands are set. This command will perform the whole optimization flow and finally show the results. This command needs no input argument. Please type “optimize”.

Optional command:

(1) set_RSM_stepsize <step size>

User can use this command to adjust the length of step vector. If user wants to perform more steps in steepest ascent, set the step size to be less than 1. The size will be directly multiplied with step vector. The input argument should be type **double**.

(2) set_RSM_design_radius <radius size>

User can use this command to adjust the design radius. If user wants a smaller initial design radius, set the radius size to be less than 1. The size will be directly multiplied with design radius. The input argument should be type **double**.

(3) set_RSM_step_shrink_factor <factor>

After an iteration of first-order model fitting and steepest ascent, the step will be shrinked by 20% due to the smaller design radius. User can adjust the shrink factor by using this command. The input argument should be type **double**.

(4) set_RSM_radius_shrink_factor <factor>

After an iteration of first-order model fitting and steepest ascent, the radius will be shrinked by 20%. User can adjust the shrink factor by using this command. The input argument should be type **double**.

(5) set_new_RSM_points_per_iteration <number>

Before performing model fitting, new RSM points are needed to fit the model. Default new RSM points per iteration is 6. User can adjust the number by this command. The input argument should be type **unsigned**.

(6) set_stop_criterion <criterion>

The iteration of first-order model fitting and steepest-ascent will be terminated if the yield cannot be improved by more than 1% of current best yield. User can adjust the value by this command. If user wants to set the criterion to be 2%, please type "set_stop_criterion 0.02". The input argument should be type **double**.

(7) set_number_of_correlation_groups <number of groups to be added>

If there are correlation groups to be added, user should use this command to set how many groups will be added. If user wants to set 6 groups, please type "set_number_of_correlation_groups 6". The input argument should be type **unsigned**.

(8) add_positive_part_to_groups <group number> <variable index>

After setting the number of correlation groups, user can use this command to add positive part to groups. User have to specify the group number and the variable index. The group number should be from 0 to (number of groups – 1). The variable index should be from 1 to (number of variables).

(9) add_negative_part_to_groups   <group number>   <variable index>

After setting the number of correlation groups, user can use this command to add positive part to groups. User have to specify the group number and the variable index. The group number should be from 0 to (number of groups – 1). The variable index should be from 1 to (number of variables).

Example Shell Script

```
// required commands
read_spice    testcases/ckt/aSi/case14T.sp
set_technology 1
set_Vth_variation_range    1.8    2.2
set_Vth_percentage 2 4 10 16 18 18 16 10 4 2
set_monte_carlo_number 1000
set_intradie_Vth_sigma 0.5
set_operation_time 10000
set_transistor_parameter_range KN 0.74 1.08
set_specs gain 10 inf
set_specs phaseMargin 55 180
set_optimization_individual_constraint 0.00001 0.0005
```

```
set_optimization_total_constraint 0.01
set_number_of_levels 2

# optional commands
set_number_of_correlation_groups 2
add_positive_part_to_groups 0 1
add_negative_part_to_groups 0 4
add_positive_part_to_groups 1 6
add_positive_part_to_groups 1 8
set_RSM_design_radius 1
set_RSM_rangesize 1
set_RSM_step_shrink_factor 0.8
set_RSM_radius_shrink_factor 0.8
set_new_RSM_points_per_iteration 6
set_stop_criterion 0.01
// to perform optimization, must have
optimize
```