

**An offset auto-calibration technique with cost-effective implementation for
comparator and operational amplifier**

by

Xinyu Gong

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering (Very Large Scale Integration)

Program of Study Committee:
Degang Chen, Major Professor
Cheng Huang
Doug Jacobson

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this thesis. The Graduate College will ensure this thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

Copyright © Xinyu Gong, 2019. All rights reserved.

DEDICATION

To my family.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
LIST OF TABLES	vii
NOMENCLATURE AND ACRONYM	viii
ACKNOWLEDGMENTS	ix
ABSTRACT	x
CHAPTER 1. INTRODUCTION	1
1.1 Comparator Concept	1
1.2 Applications of Comparators & Influence of Offset Voltage	1
1.3 Trimming Method	4
1.4 Motivation	8
CHAPTER 2. SELF-TRIMMED COMPARATOR	9
2.1 Pre-amplifier	9
2.2 Hysteresis	10
2.2.1 Definition	10
2.2.2 First Hysteresis: Schmitt Trigger	11
2.2.3 Second Hysteresis: Intended Resistor Mismatch	12
2.3 Offset Characterization	13
2.3.1 Overview	13
2.3.2 Mismatch in Circuits	14
2.3.2.1 Device mismatch	14
2.3.2.2 Offset estimation with tool	15
2.3.3 Offset Analysis	17
2.3.3.1 Random Offset in differential pair	19
2.3.3.2 Random Offset in current mirror	20
2.3.3.3 Systematic Offset in current mirror	21
2.3.3.4 Systematic offset in a single-ended output stage	22
2.4 Offset Trimming	23
2.4.1 Trimming Algorithm	23
2.4.1.1 Linear search	23
2.4.1.2 Binary search	24
2.4.1.3 Newton's search	24
2.4.2 Auto-calibration Block	26
2.4.3 Transistor Size Table	27
2.4.4 Simulation Result	28
CHAPTER 3. TRIMMING APPLICATION WITH MATLAB AND CADENCE	30
3.1 MATLAB for Offset Trimming	30

CHAPTER 4. INPUT PAIRS TRIMMING SWITCH COMPARISON	34
4.1 Scheme Comparisons	35
4.1.1 Linearity	35
4.1.2 Trimming Range.....	39
4.1.3 GBW.....	39
4.1.4 Area	40
4.2 Simulation Results.....	40
4.2.1 Trimming Curve	40
4.2.2 Post Trimming Offset with Binary Trimming Algorithm.....	42
CHAPTER 5. OP AMP OFFSET TRIMMING WITH BINARY SEARCH.....	48
5.1 Op Amp and trimming block structure.....	48
5.2 Op Amp trimming analysis.....	49
5.3 Transistor size summary	52
5.4 Simulation results	53
CHAPTER 6. CONCLUSION.....	55
REFERENCES	56
APPENDIX A. BINARY SEARCH MATLAB CODE.....	59
APPENDIX B. NEWTON’S SEARCH MATLAB CODE.....	65

LIST OF FIGURES

Figure 1.1 Comparator symbol (a) and transfer function (b).....	1
Figure 1.2 Comparator in SAR ADC.....	2
Figure 1.3 Comparator in DC-DC converter	3
Figure 1.4 Comparator in LED driver.....	3
Figure 1.5 Autozeroing technique.....	4
Figure 1.6 PXI Instrumentation Platform (ATE).....	5
Figure 1.7 Kujik bandgap with tunable resistor.....	7
Figure 2.1 Open-loop comparator.....	9
Figure 2.2 Hysteresis of the comparator	10
Figure 2.3 Schmitt trigger symbol	11
Figure 2.4 Schmitt trigger circuit.....	11
Figure 2.5 First stage of preamplifier	13
Figure 2.6 Manufacturing breakdown.....	14
Figure 2.7 MC simulation setup in cadence.....	17
Figure 2.8 Gaussian distribution of V_{th}	17
Figure 2.9 Typical first stage of pre-amplifier.....	19
Figure 2.10 Common NMOS current mirror	20
Figure 2.11 I_{ds} vs V_{ds} with channel length modulation.....	21
Figure 2.12 Resistance between supply lines	22
Figure 2.13 Second and third stage of pre-amplifier	22
Figure 2.14 Pre-amplifier stage & Input pair scheme.....	25
Figure 2.15 Offset auto-calibration.....	26
Figure 2.16 Comparator offset sigma value.....	28

Figure 2.17 Trimming range vs. trimming bits	28
Figure 3.1 Save OCEAN script from ADE L	30
Figure 3.2 Resolution error in binary search	32
Figure 4.1 Control logic and trimming switch structures: (a) switch control logic, (b) drain switch, (c) gate switch, (d) source switch, (e) split-source switch	34
Figure 4.2 Testbench for gm calculation (a) single transistor with size (W/L), (b) split source, I_{dc} biased, (c) split source, $(I_{dc} + \Delta I)$ biased	37
Figure 4.3 Offset trimming curve comparison between five types of switches.....	41
Figure 4.4 Effective trimming range.....	42
Figure 4.5 Before trimming offset generation	43
Figure 4.6 After trimming offset for (a) CGS; (b) CDS; (c) BSS; (d) SSS; (e) CSS	43
Figure 5.1 Op Amp structure	48
Figure 5.2 Op Amp with trimming blocks (a) Trim A (b) Trim B	49
Figure 5.3 Op Amp trimming on cascode stage	50
Figure 5.4 Op Amp trimming range vs. trimming bits	53
Figure 5.5 Op Amp before and after-trimming offset voltage histogram.....	53

LIST OF TABLES

Table 2.1 Comparator transistor size	27
Table 2.2 PVT simulation of comparator	29
Table 4.1 Nominal case different trimming schemes parameters	41
Table 4.2 After trimming offset voltage summary	45
Table 4.3 After-cancellation offset compared with other comparator circuits	46
Table 5.1 Op Amp transistor size	52

NOMENCLATURE AND ACRONYM

Op Amp	Operational Amplifier
CMRR	Common Mode Rejection Ratio
GBW	Gain-bandwidth Product
ADC	Analog to Digital Converter
ATE	Automatic Test Equipment
MC	Monte Carlo
CDS	Constant Size Drain Switch
CSS	Constant Size Source Switch
CGS	Constant Size Gate Switch
BSS	Binary-weighted source switch
SSS	Constant size split switch
TR	Trimming Range
PWM	Pulse Width Modulation
ATE	Automatic Test Platform
PDK	Process Design Kit
ADE L	Analog Design Environment L

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Degang Chen, and my committee members, Cheng Huang and Doug Jacobson, for their guidance and support throughout the course of this research.

I would like to thank my parents, for educating me and helping me in my life.

I would like to thank my mentors, who gave me lots of inspiration, when I was doing internship at Texas Instruments.

In addition, I would also like to thank my friends, colleagues, and department faculty and staff for making my time at Iowa State University a wonderful experience, and without whom this thesis would not have been possible.

ABSTRACT

Comparators are one of the most fundamental building blocks in all electronic systems involving analog and digital information. A comparator's performance, or the accuracy of its output, is determined by the comparator's offset voltage, which includes random offset and systematic offset. To guarantee the overall performance of an entire electronic system, offset-trimming techniques are often necessary to reduce inaccuracy. This study analyzes the offset errors in a representative comparator structure and describes an auto-calibration technique to systematically and significantly reducing the offset. The auto-calibration technique involves trimming of the comparator input transistor pair. Various trimming-switch structures are considered and compared, such as constant-sized drain switch (CDS), constant-sized gate switch (CGS), constant-sized source switch (CSS), binary-weighted source switch (BSS), and constant size split-source switch (SSS). The comparator and the offset auto-calibration circuits are designed using the GlobalFoundry 0.13 μm process. Then an offset trimming algorithm, which is written on MATLAB, is applied to these circuits. Afterwards, the results are collected and analyzed. A comparison of linearity and trimming range (TR) achieved with different trimming switch structures is performed to demonstrate advantages and disadvantages of each switch scheme. The results are also plotted in a histogram to show the normal distribution of each scheme. Finally, offset cancellation technique is implemented in an operational amplifier (Op Amp) circuit with further analysis and comparison to prove the methodology.

CHAPTER 1. INTRODUCTION

1.1 Comparator Concept

A comparator is an amplifier used in an open-loop configuration. The comparator's major function is to compare two analog current or voltage signals, one of which is typically a reference signal, and then to generate a binary output signal showing which one is larger. Basic comparator symbols and transfer functions are shown in Figure 1.1, and the idealized function is given in Eq. (1).

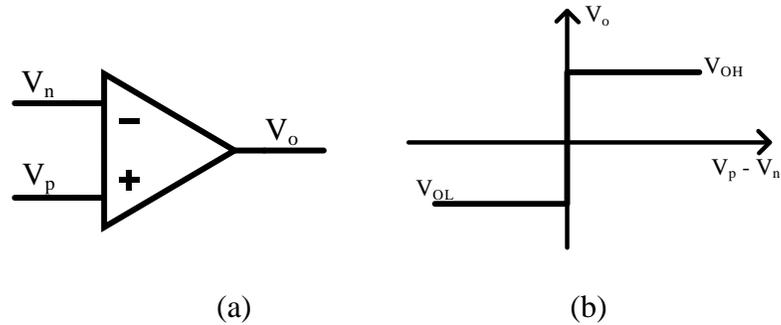


Figure 1.1 *Comparator symbol (a) and transfer function (b)*

$$V_o = \begin{cases} V_{OH}, & V_p > V_n \\ V_{OL}, & V_p < V_n \end{cases} \quad (1)$$

1.2 Applications of Comparators & Influence of Offset Voltage

Among today's circuit structures, low-offset comparators are widely used, and they play an extremely significant role in applications such as analog-to-digital converters (ADC), switching converters, Light-emitting diode (LED) drivers, and level shifters. The input offset of a comparator is the input voltage at which its output changes from one logic level to the other [1]. Ideally, the offset value is zero, but both random and systematic offsets exist in practical comparators because of device mismatches and inherently unbalanced architecture.

Offset voltage is erratic and can vary randomly between one circuit to another, even if they are produced from the same design [9]. In addition, a high offset value can negatively affect the functionality of circuit application. One of major error sources in flash ADCs and successive-approximation ADCs is comparator offset, including offset errors associated with pre-amplifiers, comparators, and resistive references, shown in Figure 1.2. Such offset mismatches could generate wrong comparison results and introduce detrimental nonlinearity to ADC accuracy [2], [3].

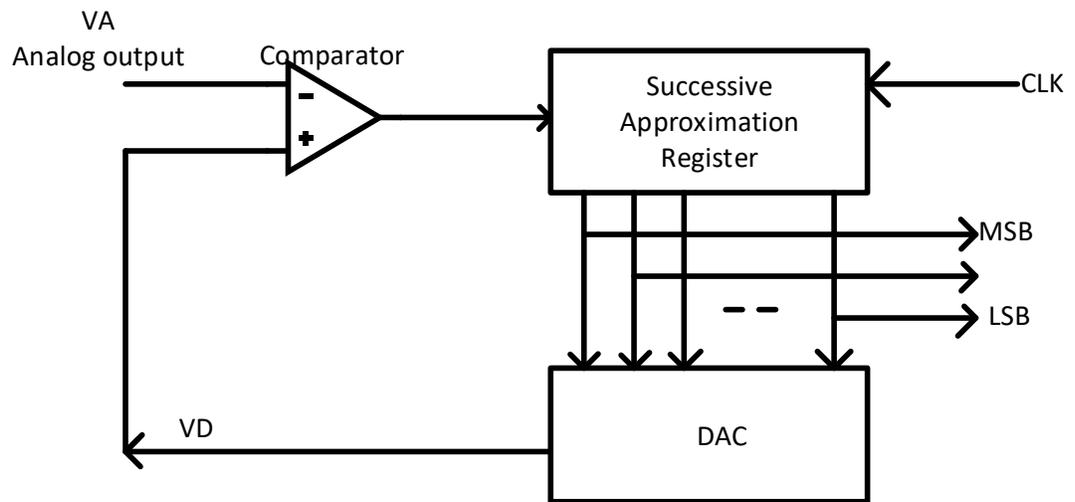


Figure 1.2 *Comparator in SAR ADC*

In a switching regulator, the comparators are widely used in a pulse-width modulation (PWM) control loop, as shown in Figure 1.3, to compare the output of an error amplifier with a fixed ramp signal to determine the instant result stored in an RS latch, thus affecting the duty cycle [4]. A large offset would trigger reset or set signal for RS latch, resulting in a wrong decision. Wrong signal will turn on high side FET or low side FET in a wrong sequence and the switching converter will have glitches. For an LED driver, show

in Figure 1.4, comparator offset could also lead to incorrect control of the voltage across the LED and cause it to become unstable and exhibit blinking issues.

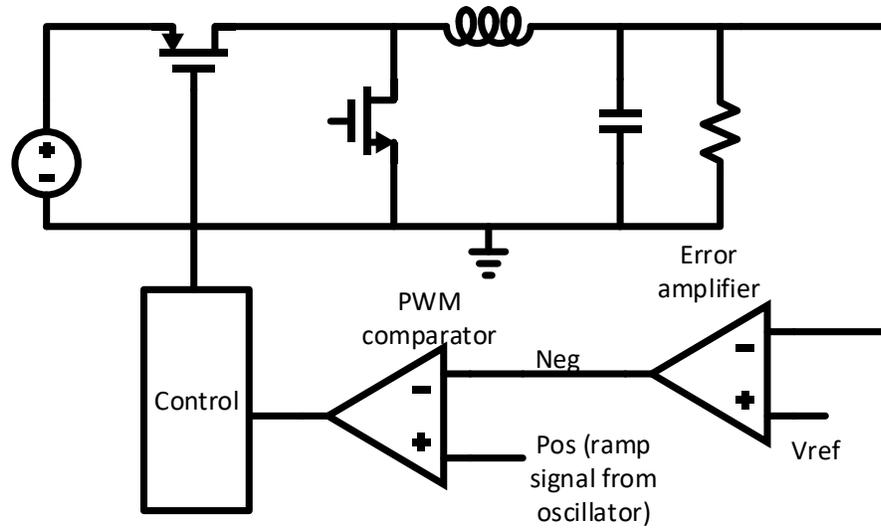


Figure 1.3 *Comparator in DC-DC converter*

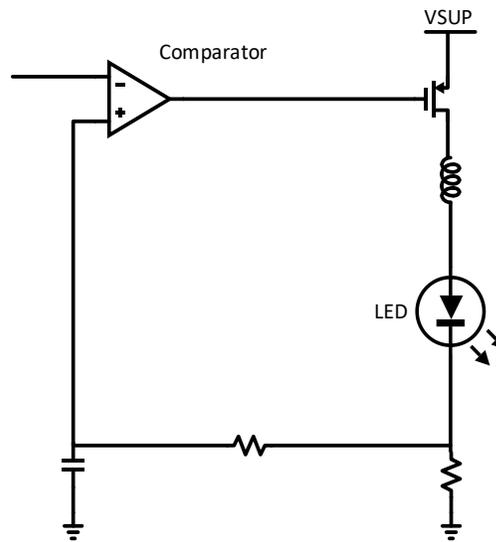


Figure 1.4 *Comparator in LED driver*

Comparators are also widely used in a variety of industrial fields to generate time-sequence signals. When complicated electronic devices are powered on, the supply voltage is the first signal to be turned on, followed by subsequent signals. How and when other

signals should be generated has become a critical issue. In such situations, comparators can be used to create “ready” signals to time the sequencing. Similarly, if the ready signal is mis-triggered by large offset voltages, the power on sequence will lead to mistaken decisions and consequently disordered time sequences.

1.3 Trimming Method

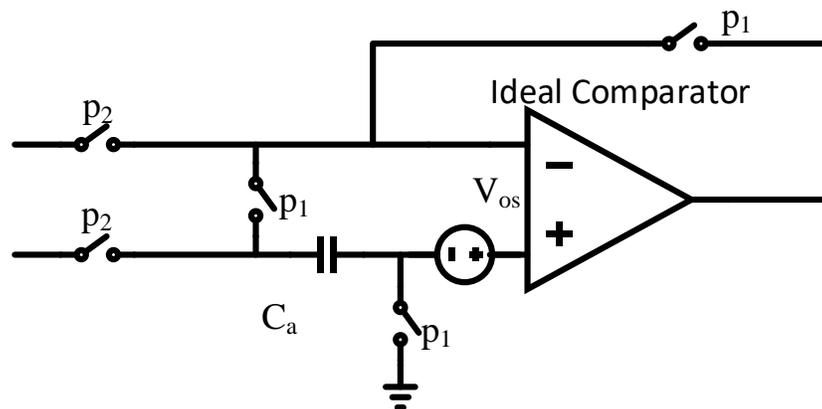


Figure 1.5 *Autozeroing technique*

To avoid such failures, usually, increasing the sizes of comparator input pairs will give more accurate matching, resulting in smaller offsets and larger transconductances (g_m), but it will increase the cost of area and power consumption. While systematic offset can be greatly reduced by precise design, unpredictable random offsets still exist [6]. Autozeroing techniques, shown in Figure 1.5, are extensively used to measure offset voltages and cancel them out using switched-capacitor circuits [5], in which an offset value is stored on capacitor C_a during a p_1 phase and cancelled on a p_2 phase. So the ideal output remains unaffected by the offset value.

An alternative offset cancellation technique, called chopping, which modulates the offset up into frequencies beyond the signal band and filters it out, has been introduced in

analog circuit applications [7]. Both methods may suffer from charge injection issues caused by switched capacitor circuits [5]. Besides, clock signals are necessary along with switching capacitor circuits to synchronize the switching phase. The generation of clock frequency will require more functional blocks such as oscillators, phase-locked loop (PLL), which makes control systems more complicated and more expensive. The general underlying concept of these two methods is to position elements around the amplifier so that offset voltages and noise can be treated differently than signals.

It is also possible to cancel offset voltages by injecting an additional DC signal that opposes the offset. For example, during wafer fabrication, post-fabrication laser trimming on an automatic test equipment (ATE) platform can effectively correct process shifts that change the device's electrical parameters [8]. Figure 1.6 shows the PCI eXtensions for Instrumentation (PXI) Instrumentation Platform whose operation is fully automatic after a programmable input setup.



Figure 1.6 *PXI Instrumentation Platform (ATE)*

During the design process, offset trimming results is usually controlled by changing passive components. Resistor trimming is the most common method used to achieve an ideal offset value because it can easily be designed as a straightforward topology. For example, a bandgap reference is the most important cell in circuit start-up blocks, because it generates all voltage references in a circuit system. The reference voltage (usually 1.25V) must be perfectly exact in a wide standard deviation range. Without such trimming, the

temperature coefficient of the bandgap voltage may change the reference voltage too much in both hot and cold temperatures. In order to get a $\pm 1\%$ accuracy bandgap voltage over six sigma range, trimming is necessary. Figure 1.7 shows a bandgap with tunable trimming resistor R_2 , and reference voltage expression is given in Eq. (2). In a one-time trimming process, the first step is to find the best temperature coefficient code. The method of choosing this code is to find which code can result in the flattest output voltage curve over application temperature ranges. Then the bandgap voltage will be the voltage on that curve at room temperature. Combined with resistive gain boost, which does not affect temperature coefficient, 1.25V reference voltage can be achieved. After running process-variation simulations, the maximum and minimum output voltage for working temperature range is limited by previously chosen temperature coefficient code. Ideally, a voltage with 1% percent change over 1.25V is a zero-temperature coefficient one. As for analog implementation of trimming circuits, all trimmable resistors will be controlled by logic circuits, and these logic circuits will be controlled by registers. Registers can be written with different values on an external pin. During the testing process, test engineers will use the PXI instrumentation platform with automatic setting mentioned above to choose different register codes for obtaining the best temperature coefficient curve.

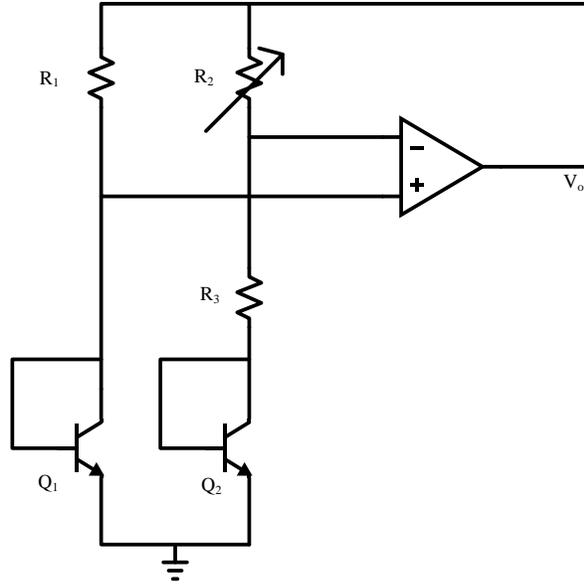


Figure 1.7 *Kuijk bandgap with tunable resistor*

$$\begin{aligned}
 V_{out} &= V_{BE2} + \frac{V_T \ln n}{R_3} (R_3 + R_2) \\
 &= V_{BE2} + (V_T \ln n) \left(1 + \frac{R_2}{R_3}\right)
 \end{aligned} \tag{2}$$

Even though the trimming can be done on automation bench, such a testing process would consume a great deal of labor and become very time-consuming. Because of this and increased demand for chip-trimming, portions of ATE based trimming have gradually moved to on-chip design [8]. An on-chip self-trimming approach auto-calibration is attractive because the trimming processes can be performed inside circuits with proper clock signals. This will not need test engineers to operate offset trimming outside, so that it can greatly decrease labor cost. Besides this, on-chip trimming is also easier to perform without inserting probes into wafer or burning fuse-links, and it is also applicable to adjustment of many on-chip parameters values (offset, voltage, current, resistance, capacitance, etc.) [8]. Its biggest disadvantage is that it is always more complicated than

conventional trimming, implying necessity of more control circuitries. Moreover, more registers mean that larger die area will be needed, and the total die size can be much larger if multiple trimming processes are implemented.

1.4 Motivation

As discussed earlier, a comparator is a particularly important structure in almost all microelectronic circuits, and offset voltage is one of the most crucial specifications in evaluating comparator performance. On-chip self-trimming method is becoming increasingly popular due to its advantages. A detailed on-chip trimming approach to comparator like circuits should be investigated and discussed. Different types of switches for trimming transistors can lead to different performances. Comparison between these switches is also necessary.

Furthermore, circuits like operational amplifier (Op Amp) can also incorporate an on-chip self-trimming method because the similarity between Op Amps and comparators. Then the method of translating it from a comparator application to an Op Amp application should be explained.

CHAPTER 2. SELF-TRIMMED COMPARATOR

Open-loop comparators are operational amplifiers (Op Amps) without compensation to which pre-amplifiers can be added to amplify the signal for obtaining higher resolution and reducing kickback effects [1]. For a typical comparator, the preamplifier will form the first stage. A Schmitt trigger will be the second stage, and it is

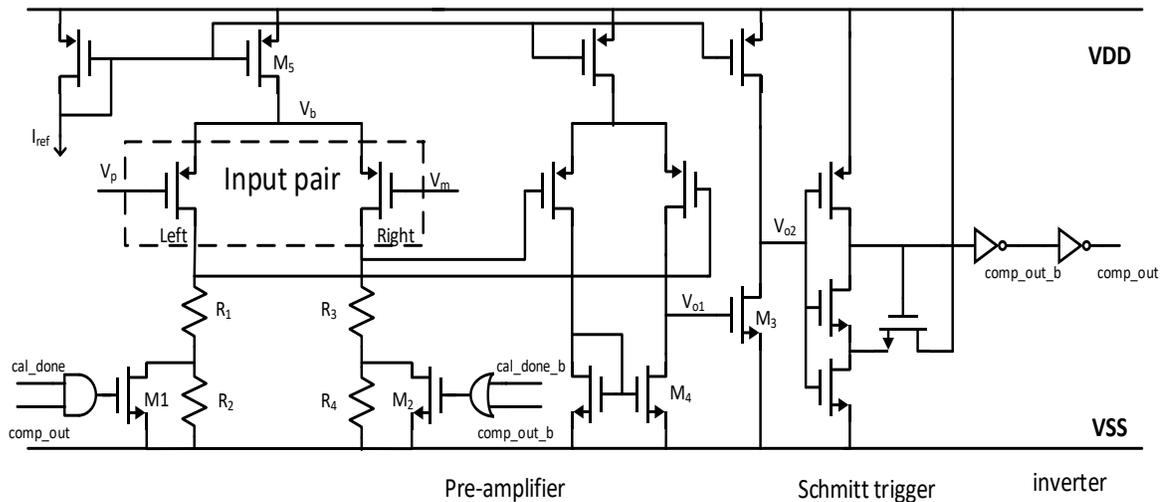


Figure 2.1 *Open-loop comparator*

followed by a third stage inverter stage. As shown in Figure 2.1, this comparator is a specific designed comparator for this thesis work, and the design details will be introduced. The offset analysis based on this comparator will also be performed.

2.1 Pre-amplifier

Comparator stages can be divided into pre-amplifier stages, Schmitt trigger and inverter stages. A preamplifier, used to increase comparator speed, can be built to amplify the input signal up to a level large enough to trigger the inverter stage to make a comparator flip.

Since we know that the gain-bandwidth product of a comparator is usually constant, preamplifier design criteria can be considered a tradeoff between speed and bandwidth. For the design considered here, the comparator has three pre-amplifier stages, each with gain of about 20dB.

2.2 Hysteresis

2.2.1 Definition

Hysteresis will usually be added after a pre-amplifier stage to reduce the noise effect caused by the final pre-amplifier stage, and combat mis-triggering that could lead to an incorrect comparison result.

Hysteresis is the comparator quality whereby the input threshold changes as a function of input (or output) level [6]. The original idea of creating hysteresis is to generate positive feedback to change the threshold value once it has been exceeded. For example, once a signal changes from low to high, the rising threshold value is decreased so that the existing high signal will not turn back to be low as if it lies within the hysteresis range, as shown in Figure 2.2. Hysteresis can always be added with a Schmitt trigger to reduce its sensitivity to noise.

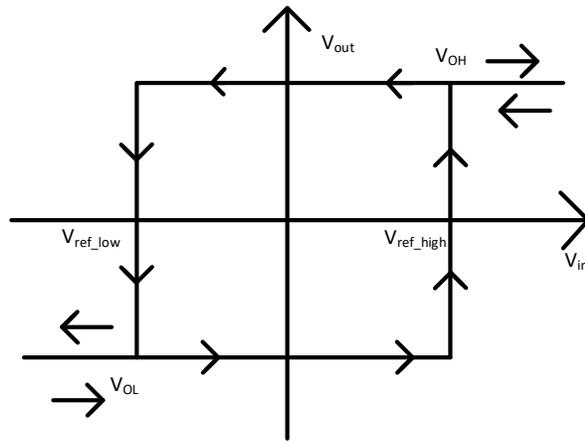


Figure 2.2 *Hysteresis of the comparator*

Different rising and falling threshold values can ensure that, once the comparator is triggered, it will not be affected by small error sources such as noise. After adding hysteresis window, if the noise is no large than the hysteresis range, then the comparator will not be mis-triggered.

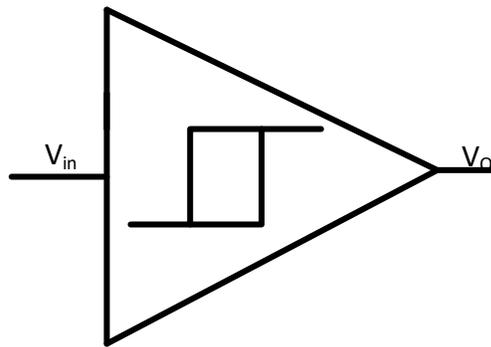


Figure 2.3 *Schmitt trigger symbol*

2.2.2 First Hysteresis: Schmitt Trigger

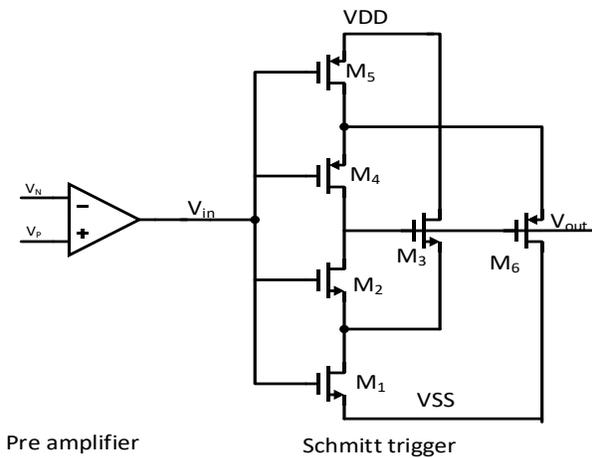


Figure 2.4 *Schmitt trigger circuit*

A Schmitt trigger is a standard block widely used to generate hysteresis. The symbol for a Schmitt trigger is shown in Figure 2.3. It is similar to an inverter except with more precisely defined trigger points, as shown in Figure 2.4.

Circuit operation is as follows: If we assume that V_{in} is low, V_{out} will be high, and M_3 , M_4 and M_5 will so be on, while M_1 , M_2 and M_6 will be off. As V_{in} begins increasing from zero toward VDD, M_2 gradually begins to turn on, resulting in closure of M_3 . Because of the M_3 drain connection to VDD, positive feedback causes the V_{gs} of M_2 to become larger, and M_2 is further turned on by this positive feedback, and eventually both M_1 and M_2 are on and the output becomes zero. Similarly, the PMOS side creates hysteresis when V_{in} transitions from high to low. In this comparator, only one side of the Schmitt trigger is implemented because we want the comparator to work if V_{in} is larger than V_{ref} , meaning that there is no hysteresis as V_{in} transitions from low to high.

2.2.3 Second Hysteresis: Intended Resistor Mismatch

Another type of hysteresis is implemented in the first stage of the pre-amplifier of Figure 2.5. M_1 and M_2 work as switches and $R_1 + R_2 = R_3$. When $V_p < V_m$, R_4 is shortened by M_2 and R_1 , R_2 , R_3 are connected, and the circuit becomes balanced. When $V_p > V_m$, R_2 is shortened by M_1 , $R_3 + R_4 > R_1$, and so hysteresis is created. Note that the size of M_2 must be large for minimizing mismatch induced by its R_{on} . During comparator calibration, hysteresis should be disabled or else the calibration system will treat it like offset. The control logic is energized by a `cal_done` signal and logic gates. When the comparator is in its trimming phase, `cal_done = 0`, `cal_done_b = 1`, meaning M_1 is off, M_2 is on, and the hysteresis is disabled.

2.3.2 Mismatch in Circuits

Pelgrom parameters [18] are usually used to describe mismatches of the μC_{ox} and the threshold voltage V_{th} . Pelgrom parameters are process parameters which are represented by $A_{V_{th}}$ and A_{WL} .

Process variation causes mismatch occurring in variation during factory transistor manufacture. Length, width, oxide thickness, and other variables vary for each device. As devices become smaller and smaller, even though the absolute error range may not change, mismatch percentage increases.

2.3.2.1 Device mismatch

Both transistor-level design and layout style can affect total mismatch. Transistor-level design is most vulnerable to parametric yield issues caused by process variations during manufacturing. Although a minor change may only lead to a small variation for a single transistor, considerable variation can occur because there are hundreds of millions of transistors in one system. Temperature and humidity in areas have big effects to wafers' manufacturing. Variation in key parameters such as gate oxide thickness and doping area can also change transistor performance.

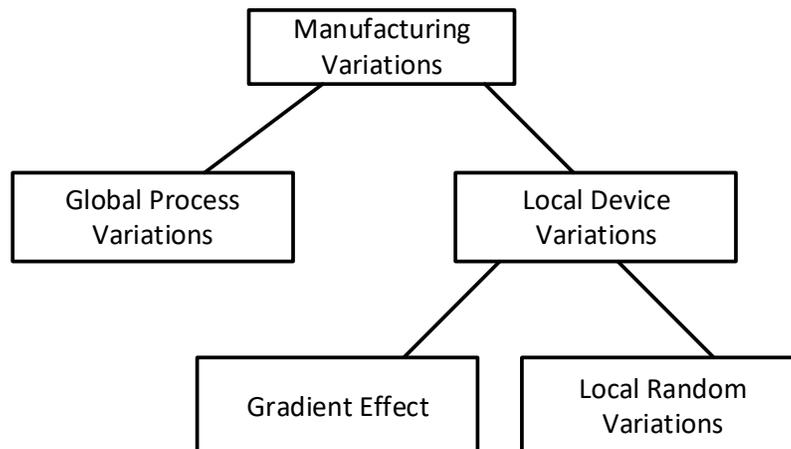


Figure 2.6 *Manufacturing breakdown*

All the factors described are known to cause local device variation. In Figure 2.6 [19], manufacturing variations are categorized into two main parts: global process variation from wafer to wafer, and local device variation that includes gradient effects and random local variations [19]. As mentioned earlier, layout style can also affect total mismatch. Since gradient effects are known to contribute to random mismatch, an important technique called common-centroid layout is widely used to minimize mismatch. In this work, the use of common-centroid layout is assumed, and the analysis will focus on local random variations.

2.3.2.2 Offset estimation with tool

The usual way to estimate effects of process variation is to run Simulation Program with Integrated Circuit Emphasis (SPICE) simulations in Cadence over digital process corners provided by the manufacturing foundry in a process design kit (PDK). Digital process corners, provided by the foundry, are typically determined by I_{dsat} characterization data for N and P channel transistors. Positive and negative three sigma values could be selected to represent fast and slow corners for such devices [20]. If the model is sufficiently accurate, the estimation should be able to cover all potential problems caused by global variations, although digital process corners have limitations in two respects [20]:

1. Digital corners account only for global variation, and they are represented as “slow” and “fast” in digital design context.
2. Digital corners, mainly developed for digital design do not include effects of local variation that is critical to analog design.

The fact that corners only cover global variational effects means that, for local simulation, the offset should not vary too much over the corners, and it is important to run other simulations to estimate the real offset more accurately.

Monte Carlo (MC) simulation is the most dependable and commonly used method for measuring the effect of process variations. Although MC simulation includes different statistical methods, simulation results always provide us with an accurate estimation of yield because the basic idea underlying MC analysis is to simulate many random statistical samples based on statistical process models. [20] In the circuit design tool Cadence Virtuoso, MC simulations are usually performed using the circuit simulator SPECTRE that can use a MC algorithm to generate a set of circuit variants for determining randomness variation. If the models are accurate enough, simulation results can represent real local variations and the sigma value, which is an important parameter for estimating yield. Note that, in Cadence MC simulation setup, both process and mismatch can be included, but only mismatch should be considered for estimation of local variation. For corner setup in MC simulation, use of normal corners and 100 runs are usually enough. Saving mismatch data for performance analysis is recommended and the “save data to allow family plots” option is necessary for plotting normal distribution graphs, as shown in Figure 2.7.

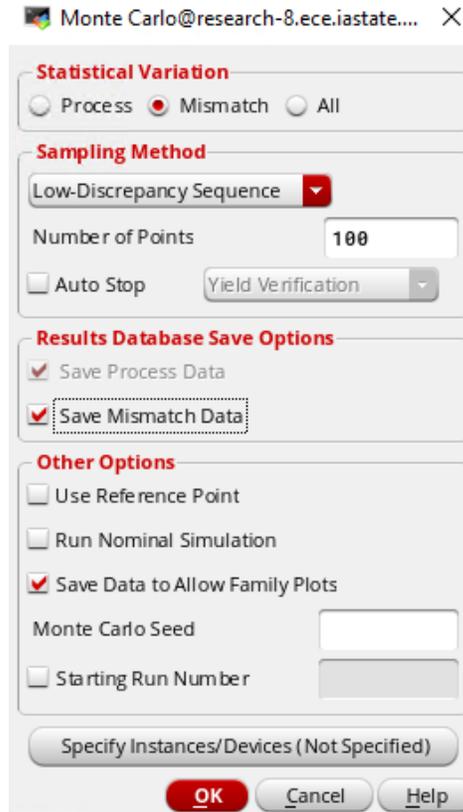


Figure 2.7 MC simulation setup in cadence

2.3.3 Offset Analysis

Random offset is caused by mismatches among equally placed transistors. The parameters V_{th} (threshold value) and μC_{ox} are measured, and the resulting distribution of V_{th} is shown in Figure 2.8 [21].

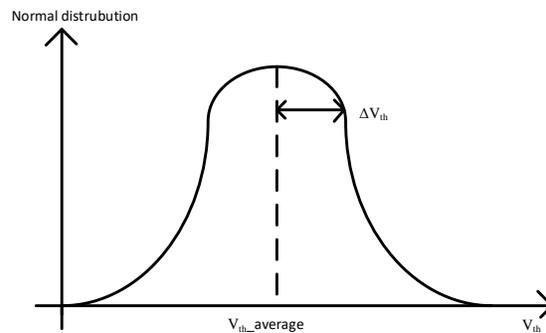


Figure 2.8 Gaussian distribution of V_{th}

Gaussian distribution is usually associated with an average value and sigma value, and models showed that this sigma value was inversely proportional to the square root of the product of width and length (area). $A_{V_{th}}$ itself depends on the process technology, so it is a constant for a given process [21].

$$I_{ds} = K \frac{W}{L} (V_{gs} - V_{th})^2 \quad (3)$$

$$\sigma_{\Delta V_{th}} = \frac{A_{V_{th}}}{\sqrt{WL}} \quad (4)$$

$$A_{V_{th}} \sim t_{ox} \sqrt[4]{N_B} \quad (5)$$

Parameter N_B is the doping level of substrate, t_{ox} is the oxide thickness. For smaller channel length L , the doping level increases but t_{ox} decreases. In this comparator design, in order to minimize the offset caused by V_{th} shift, tying source body together is used for PMOS transistors. Such that the body effect can be eliminated as the voltage of source and the voltage of bulk is under the same level.

Similarly, other transistor parameters following this trend can be expressed as shown in Eq. (6) and (7) [21].

$$\sigma_{\Delta K} = \frac{A_K}{\sqrt{WL}} \quad (6)$$

$$\sigma_{\Delta WL} = \frac{A_{WL}}{\sqrt{\frac{1}{W^2} + \frac{1}{L^2}}} \quad (7)$$

For example, for a PMOS transistor, the $A_{V_{th}}$ value can be about $0.005\mu\text{m}$ while the A_{WL} value is $0.02\mu\text{m}$, so transistor size clearly dominates random offset variation. It is also important to know that the Pelgrom parameters are generally higher for NMOS than for PMOS.

2.3.3.1 Random Offset in differential pair

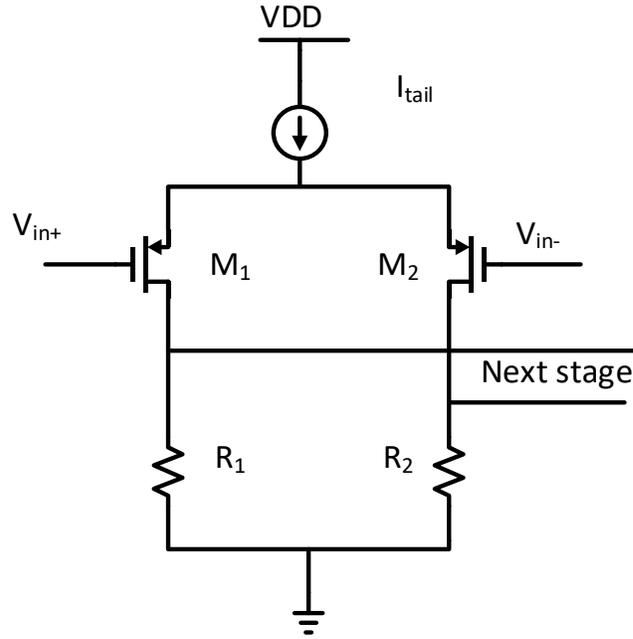


Figure 2.9 *Typical first stage of pre-amplifier*

Figure 2.9 shows the first stage of a pre-amplifier. Suppose R_1 and R_2 have mismatch such that

$$R_1 = R_2 + \Delta R \quad (8)$$

The differential gain is then

$$V_{od} = \Delta R \frac{I_{tail}}{2} \quad (9)$$

Suppose each of M_1 and M_2 has a transconductance g_m . The transconductance can transfer an offset voltage V_{os} to a current. The current equals to V_{od} divided by R_1 . This situation can be expressed as

$$V_{os} = \frac{V_{od}}{g_m * R_1} \quad (10)$$

Substituting Eq. (9) into Eq. (10), we obtain Eq. (11)

$$V_{os} = \frac{\Delta R}{R} * \frac{I_{tail}}{2g_m} \quad (11)$$

Since $\frac{g_m}{I_{ds}} = \frac{2}{V_{gs} - V_{th}}$, the result is

$$V_{os} = \frac{\Delta R}{R} * \frac{V_{gs} - V_{th}}{2} \quad (12)$$

Eq. (12) indicates that input pair transistors should be designed for high gain, meaning that the transistor has a small $V_{gs} - V_{th}$. So with respect to decreasing V_{os} , strong inversion is not recommended for input-pair biasing; only moderate inversion or weak inversion is suggested. Indeed, for weak inversion, the MOSFET behaves like a bipolar transistor whose offset voltage is dramatically smaller than that of a MOSFET in strong inversion, because it has larger area, and better g_m efficiency is internally performed [20]. Further deduction can be made, producing the result shown in Eq. (13).

$$V_{os} = \Delta V_{th} + \frac{V_{gs} - V_{th}}{2} \left(\frac{\Delta R}{R} + \frac{\Delta K}{K} + \frac{\Delta WL}{WL} \right) \quad (13)$$

Notably, these factors may occur under worst case because they are not related and cannot be summed and therefore cannot cancel one another.

2.3.3.2 Random Offset in current mirror

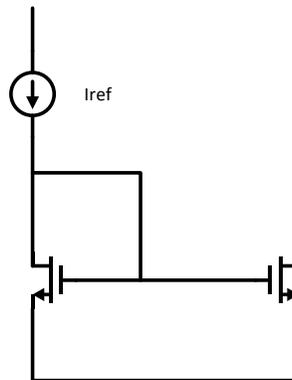


Figure 2.10 Common NMOS current mirror

The offset in a current mirror can be similarly described by

$$\frac{\Delta I_{out}}{I_{out}} = \frac{\Delta V_{th}}{(V_{gs} - V_{th})} + \frac{\Delta K}{K} + \frac{\Delta WL}{WL} \quad (14)$$

Eq. (14) tells us that, to have a good matching, the $V_{gs} - V_{th}$ value of current mirror should be large enough to minimize offset between them. Since I_{ref} remain constant, we usually increase the length of the mirror pairs to achieve larger $V_{gs} - V_{th}$.

2.3.3.3 Systematic Offset in current mirror

In addition to random offset, systematic offset is also important from a design consideration. Usually, systematic asymmetry causes systematic offset, and it can be eliminated with a careful design.

For example, in a current mirror, there are two types of systematic offset. Figure 2.11 [21] shows that even without random offset a small difference between V_{ds} will lead to a current difference, an effect called channel length modulation. The larger the channel length, the flatter the plot will be and the smaller the mismatch.

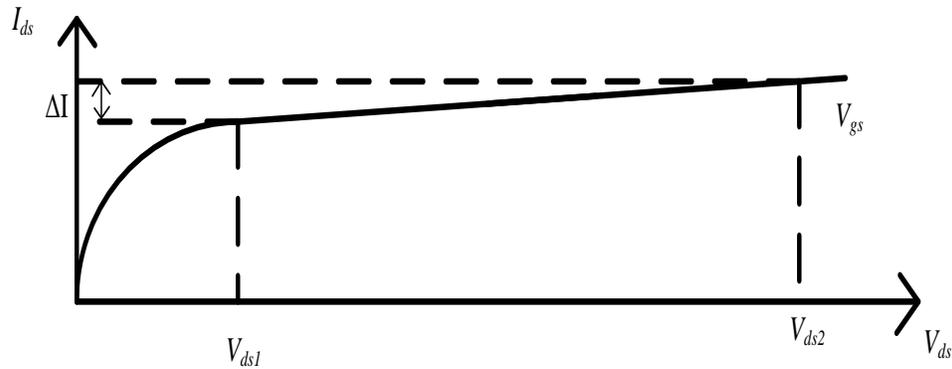


Figure 2.11 I_{ds} vs V_{ds} with channel length modulation

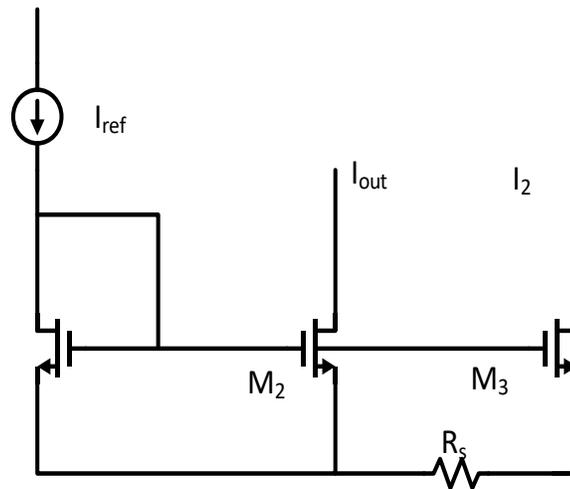


Figure 2.12 Resistance between supply lines

Also, in a real layout scheme, supply line resistance can also cause a mismatch, as shown in Figure 2.12. The voltage drop across R_s will decrease the actual value of V_{gs} of M_3 , making the actual gate to source voltage smaller. And because it occurs in every circuit, we can consider it as a systematic offset.

2.3.3.4 Systematic offset in a single-ended output stage

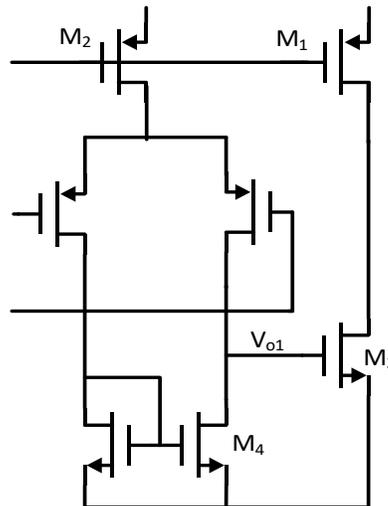


Figure 2.13 Second and third stage of pre-amplifier

The size ratio between M_3 and M_4 and that between M_1 and M_2 are important, as shown in Figure 2.13. If we have $M_2:M_1 = 2:1$, current flows through M_4 will equal current flows through M_3 . In this case, when the comparator is switching, the capacitance at V_{o1} will be charged or discharged, expressed as sink and source ability. If the size ratio between M_3 and M_4 is not 1, then the $V_{ds,sat}$ value between them will be different, so the ability to increase voltage at node V_{o1} will be different. This systematic offset can be cancelled by a proper design.

2.4 Offset Trimming

Trimming is using special methods to cancel unwanted effects. With respect to offset trimming, an intentional added offset can be used to cancel existing offset, so we must determine how to detect the original offset and cancel it.

Input pair trimming is used for comparator offset trimming, and to do this we can either control the resistor value in each first-stage branch or perform input pair trimming on the second stage, referred to as a fine trim. We call it fine trim because the offset added on the second stage will be divided by the first stage gain.

2.4.1 Trimming Algorithm

Figure 2.14 portrays an input pair comparator offset trimming scheme in which a constant size drain switch (CDS) is currently used. A trimming search algorithm can be categorized into either linear search or binary search.

2.4.1.1 Linear search

Linear search, also called sequential search, consuming most time among search methods. The method is to increase the trimming bit from one end until the comparator flips [15]. This method sweeps linearly, and while it uses simple and straightforward logic,

it wastes time because it must sweep completely from the bottom to the top to cover every bit.

2.4.1.2 Binary search

Starting from the most significant bit (MSB) toward the least significant bit (LSB), we tag the comparator output as a high or low one at the beginning. If it is high, we will try to find which bit can make the comparator output change from high to low, and vice versa. We will compare the output generated by the middle trimming bit with the target value. Then we can determine the half in which the target cannot exist and keep searching with the other half. Again, we can take the middle point to repeat the same thing until we find the LSB. This algorithm uses logic similar to that of a successive-approximation analog to digital converter [16], and it is more complicated but more time-efficient than a linear search because once the matching requirement is met, the search will stop and results will be returned. In this study, the binary search method was implemented for a trimming-bit feedback loop. Appendix A shows how it works.

2.4.1.3 Newton's search

Compared to the above mentioned two methods, Newton's search is the most direct method for determining a trimming code. If trimming curve linearity is good enough, it will be easy to predict trimming codes by calculating the offset that can be detected. For example, if the trimming range is X and total number of trimming bits is A , the total number of trimming steps is X/A . We assume a detected offset voltage of Y , then $Y/(X/A)$ can be easily calculated, and it would be the desired trimming bit. The more linear the trimming curve, the more accurate the trimming bit will be, meaning that the integral nonlinearity (INL) requirement for trimming curve must be extremely high, or else the trimming code based on the calculation would most likely fall into other step ranges. The detailed

Newton's search algorithm is described in Appendix B.

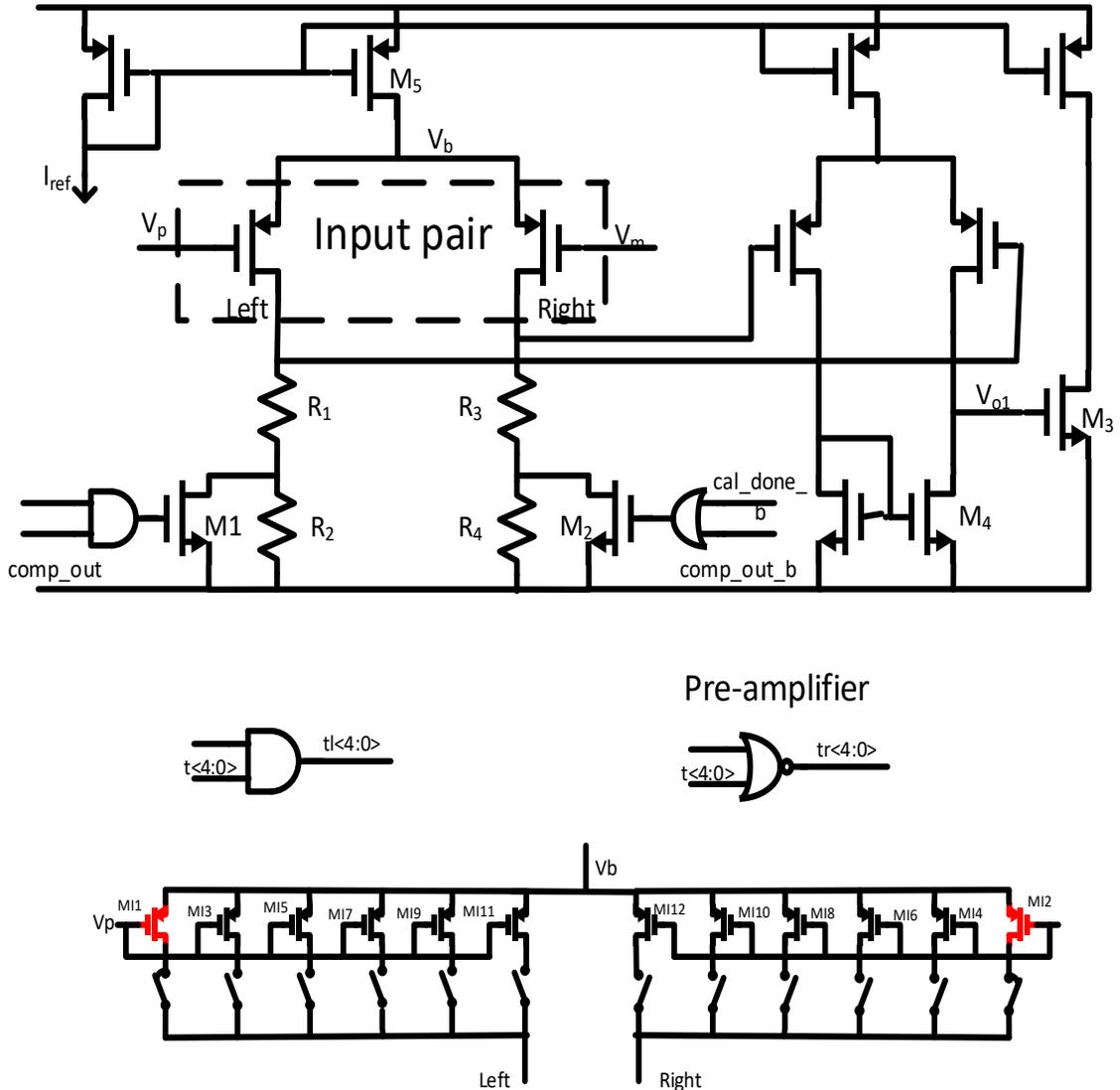


Figure 2.14 Pre-amplifier stage & Input pair scheme

Figure 2.14 shows an input pair trimming part using drain switches. Transistors in red are the main input pairs that are always connected, while the black transistors are trimming input pairs whose sizes are binary-weighted. Instead of using ten bits to control all switches, a waste of register area, six bits for controlling signal and gate logic are implemented in Figure 2.14. The MSB of the control signal is a pseudo bit for generating the actual control signal $tl<4:0>$ for the left side switches and $tr<4:0>$ for the right side

switches, i.e., if PMOS is used as a switch, when the MSB is zero, $t_{l<4:0>}$ is always zero, opening trimming transistors on the left side. While $t_{r<4:0>}$ depends on the $t_{l<4:0>}$ value, so offset is created purposefully. Before trimming, $t_{l<5:0>}$ is set to 100,000, meaning that all trimming pairs are on to have the largest size of input pairs.

2.4.2 Auto-calibration Block

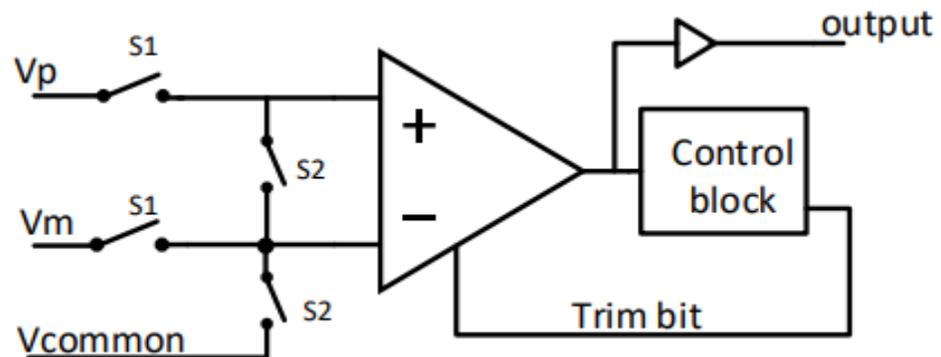


Figure 2.15 Offset auto-calibration

Figure 2.15 shows the auto-calibration block in which `cal_done` controls switch S_1 and S_2 . When `cal_done` is low, the comparator goes into calibration mode, S_1 is off, S_2 is on, and the comparator inputs are shorted to a common voltage, the desired reference voltage. When `cal_done` is high, S_1 is on, S_2 is off, and the comparator works normally, and the comparator output is sent out through a buffer. During calibration mode the control block controls the trimming algorithm written with MATLAB coding. Depending on the comparator output logic (high or low), the control block generates a new trim code after receiving the output result. A clock signal is implemented on the control block to show a suitable period for the comparator to settle down after a new trim code is applied. The trimming process can be described as: denote the first logic level, then change the trimming

bit to see at which bit the first logic level changes to the opposite value in a binary-search method. After trimming, the offset value should not be larger than the step size between two bits.

Table 2.1 *Comparator transistor size*

Transistor number	W/L (um)	Transistor number	W/L (um)
M ₁	(0.5/0.4) *2	M ₁₀	(3.6/1) *2
M ₂	(5/0.7) *20	M ₁₁	(1/1) *4
M ₃	1/1	MI _{1,2}	(1/1.8) *32
M ₄	(1/1) *4	MI _{3,4}	(0.5/2.5) *16
M ₅	(5.5/7) *16	MI _{5,6}	(0.5/2.5) *8
M ₆	5.5/7	MI _{7,8}	(0.5/2/5) *4
M ₇	(5.5/7) *8	MI _{9,10}	(0.5/2/5) *2
M ₈	5.5/7	MI _{11,12}	(0.5/2/5) *1
M ₉	(3.6/1) *2		

2.4.3 Transistor Size Table

The sizes of transistors in the comparator are listed in Table 2.1. As mentioned earlier, the sizes of M₃ and M₄ are proportional to the current flow through each branch; the transistor used for current mirror should have a large length to achieve better matching. The input of main pair sizes is large enough to minimize the offset effect due to $\Delta(WL)$, and it is also important to use large-sized main pairs to achieve small sigma variation in MC simulation. Otherwise, the trimming range must be extended by designers, and the number of trimming bits will increase a great deal. The transistors used in the trimming branches are binary-weighted sizes. Six trimming bits are used, with one sign bit and five

trimming bits, and calculation of the trimming bits is based on the MC simulation. For example, MC simulation would be run to obtain the sigma value, as shown in Figure 2.16.

Test	Output	Min	Max	Mean	Median	Std Dev
trm_explore:testbench:1	offset	-4.35m	5.75m	-193u	-350u	2.083m

Figure 2.16 *Comparator offset sigma value*

Since the six-sigma value will then be 12.6mV, in order to achieve the target (trim offset to 0.5mV in six-sigma range), $12.6\text{mV} / 0.5\text{mV} = 25.2$ bits are required for half side. Since offset can be either positive or negative, a total of about 50 decimal trimming bits is necessary, and six binary bits ($2^6 = 64$) can cover this value.

2.4.4 Simulation Result

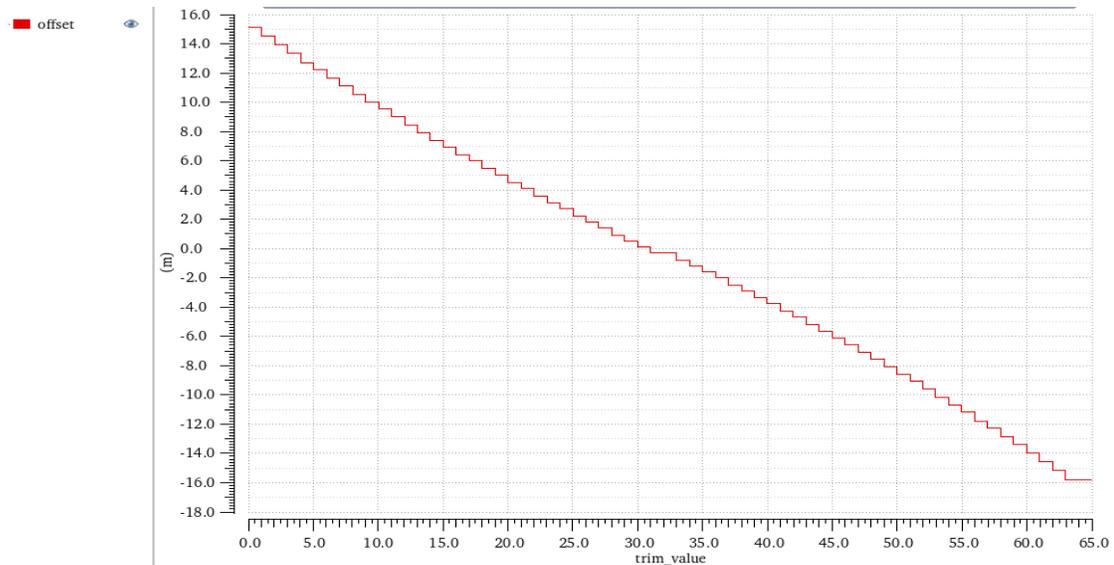


Figure 2.17 *Trimming range vs. trimming bits*

Figure 2.17 shows the simulation results, showing that the total range is enough to cancel 12.06mV with a step size less than 0.5mV. This is only an offset voltage vs. trimming bits plot, and the comparator has yet not been trimmed. If the trimming algorithm

is correct, an after-trimming offset should be expected. In addition, CDS was used for this example, and other types of switches remain to be explored.

Notice that before trimming we also must run process, voltage, and temperature (PVT) variation simulation to predict the offset voltage over corners. Ideally, there should not be too much movement because we do not have mismatch in this simulation, although we must be careful about systematic error that could result in larger offset, as mentioned before.

Table 2.2 lists the PVT simulation results. In Table 2.2, the absolute value of offset voltages moves from $10\mu\text{V}$ to $90\mu\text{V}$, which is an acceptable value. With respect to other parameters, in the nominal case, the total quiescent current is $15.5\mu\text{A}$ and the propagation delay time is 89nS . If a higher quiescent current is used, the offset without trimming would improve, mainly because the current created by the current mirror matches better. However, since we will trim it, there is no need to consume too much current as long as we make the original comparator offset trimmable. Also, this is not directly related with trimming, the topic of focus, so no further discussion is needed.

Table 2.2 *PVT simulation of comparator*

Parameter	Condition	Results			Unit
		Min	Typ	Max	
Input offset	PVT	-10	-30	-90	μV
Propagation delay	PVT	46.46	89.59	99.7	nS
Supply current low	PVT	15.04	15.29	15.38	μA
Supply current high	PVT	15.35	15.74	15.84	μA
Hysteresis	PVT	14.3	17.02	21.34	mV

CHAPTER 3. TRIMMING APPLICATION WITH MATLAB AND CADENCE

3.1 MATLAB for Offset Trimming

One method to perform offset trimming is to use MATLAB for the digital work and Cadence Virtuoso for the analog work. MATLAB can “wake up” Cadence when it is needed, and the simulation results can be returned to MATLAB for analysis.

Open Command Environment for Analysis (OCEAN) allows setting up, simulating, and analyzing circuit data without using Virtuoso Analog Design Environment L (ADE L), XL, or GXL. The basic idea would be to write MATLAB commands to execute OCEAN script in Cadence. OCEAN was written with the SKILL language, the same language used in the Cadence environment to describe circuit settings, and circuits netlist creation is based on SKILL commands. We can save SKILL script files from ADE L settings and call a script to perform the same function as ADE L.

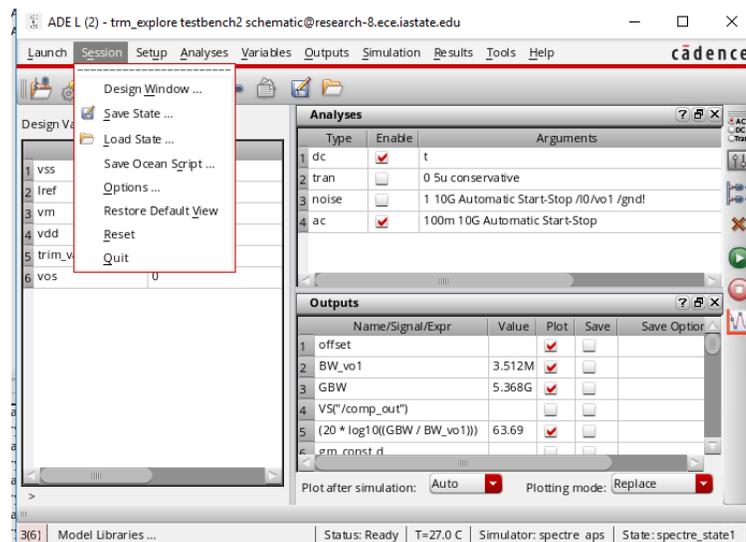


Figure 3.1 Save OCEAN script from ADE L

Now we should be able to start the OCEAN script from within MATLAB. If we have an open MATLAB window in Windows system, there is a MATLAB command DOS which will allow execution of a command in the Windows shell. From the Windows shell, plink (part of Putty) can be used to execute the Ocean script on the Linux server that runs Cadence. The MATLAB code to perform this sequence is as follows:

```
cmd='c:\Program Files"\PuTTY\plink -l xygong research-8.ece.iastate.edu'
cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=', string(vos(i)),');
cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/bin_2.ocn;')
dos(cmd);
```

Using command “strcat” in MATLAB can create the complete command to be executed in a MATLAB variable called cmd for better organization and readability. bin_2.ocn is the OCEAN file having all the settings used for offset trimming.

However, if the MATLAB code is executed , while Cadence can run the OCEAN script, the results are not returned to MATLAB, so a command should be added to Ocean Script that saves the desired Cadence results to a file to be read into MATLAB. Use the command “ocnPrint” in the OCEAN script as follows:

```
ocnPrint(?output "./bin_2_result" comp_out ?precision 15 ?numberNotation 'none
?from 1m ?to 10m)
```

In this example, bin_2_result specifies the filename to be written and comp_out is the after-trimming offset value.

Now commands must be added to the MATLAB file to read and parse the file written by Cadence. Command “fopen” can read back the result files you just saved by

```
filename = '\\h10.ece.iastate.edu\xygong\lala_offset';
fid = fopen(filename,'r');
A = fscanf(fid,'%e',[1 inf]);
t(i) = A(:,1);
```

using the command “fscanf” to read the value back and save it to a variable.

With this method, binary search trimming algorithm can be implemented for comparator offset trimming. Coding details can be found in Appendix A.

However, with this type of design, there is always a one-bit selection accuracy issue, the so-called resolution error. For example, Figure 3.2 depicts such an error case.

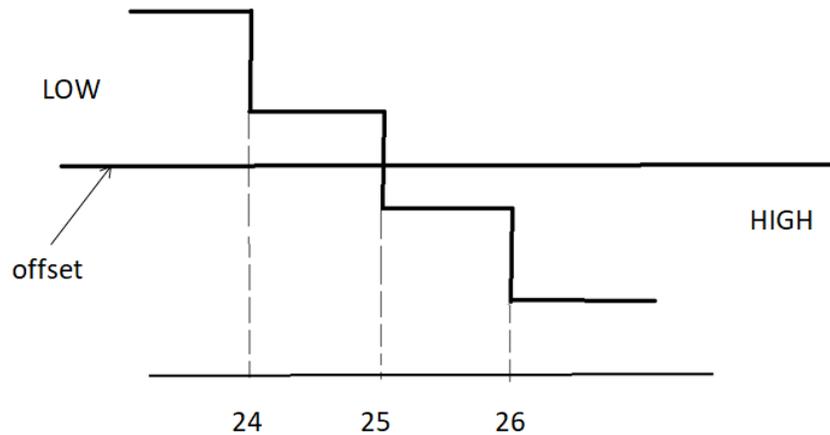


Figure 3.2 *Resolution error in binary search*

In this case, the offset is between trimming bit 24 and trimming bit 25. Bit 24 gives a low value and bit 25 gives a high value. Technically, both bits 24 and 25 are available, although our algorithm always chooses the left one (bit 24).

The Monte Carlo (MC) simulation data without adding one more bit indicates that an offset mean value much larger than 0 (deviation from origin) always exist. This is not what we want if we consistently choose left bit for all cases. In this example, we are using trimming bit 24 to 26, which means lower range of trimming curve is used. In this range, the right-side bit is always a better choice, and it will not make the left-side one too strong to deviate from the value we need.

However, when the higher range is chosen, we need not add one more bit after trimming, because, if we add one more bit, the right side will become too strong and make the offset value much smaller than original value.

CHAPTER 4. INPUT PAIRS TRIMMING SWITCH COMPARISON

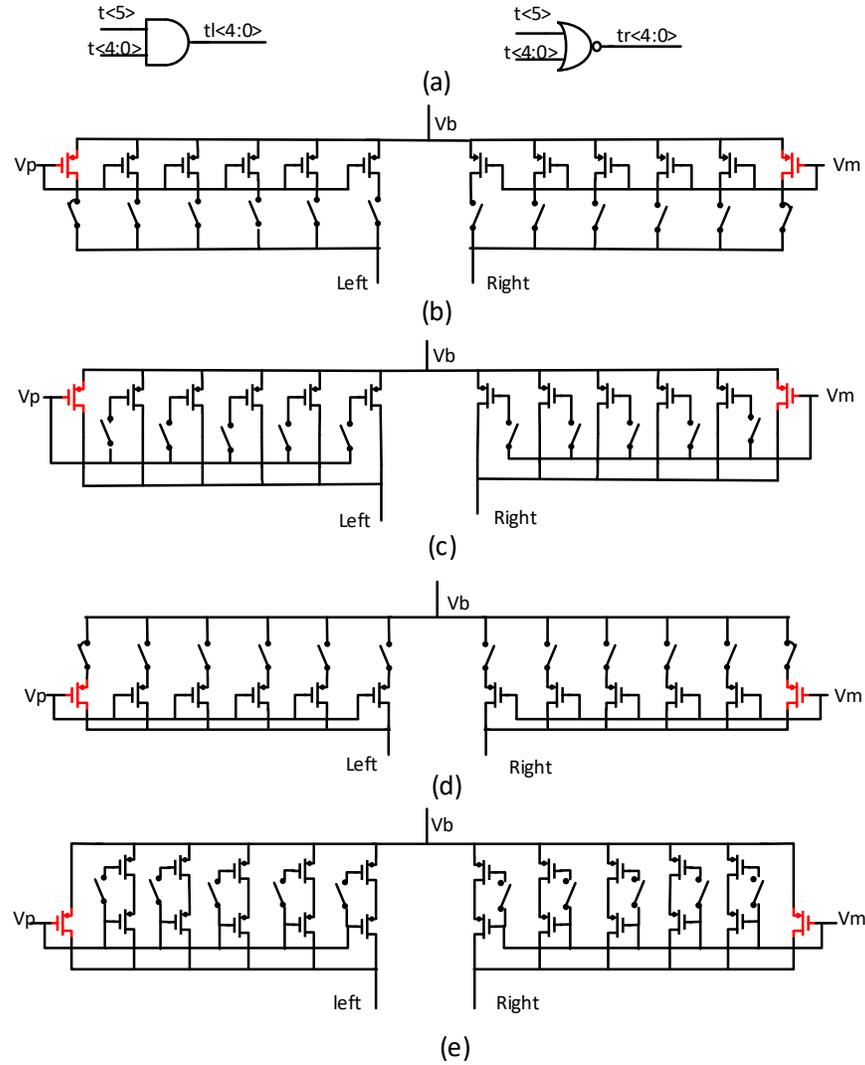


Figure 4.1 Control logic and trimming switch structures: (a) switch control logic, (b) drain switch, (c) gate switch, (d) source switch, (e) split-source switch

Four types of input pair trimming switches: constant size drain switch (CDS), constant size gate switch (CGS), source switch, and constant size split source switch (SSS) are shown in Figure 4.1. The source switch includes a constant size source switch (CSS) and a binary-weighted size source switch (BSS).

4.1 Scheme Comparisons

4.1.1 Linearity

Maximum trimming step size determines the offset resolution after trimming, and the step size of offset trimming bits is inherently determined by to what extent the g_m of the trimming bits can affect the g_m of the main input pairs. Technically, the step size should be linear and kept constant, because the trimming input pairs are binary-weighted and the main input pairs always work in saturation region. However, because of the non-linearity issue, the trimming step size may increase to a value that is larger than the required after-trimming offset voltage. The drift of voltage V_b shown in Figure 4.1 causes non-linearity. Suppose there is an offset voltage on the left side; as the offset voltage increases, to keep the same V_{gs} of input transistors, the V_b will increase so that the V_{ds} of tail transistor M_5 will decrease. Because M_5 operates in the saturation region, and it is influenced by channel length modulation effect, the tail current can be expressed as:

$$I_{ds} = \frac{1}{2} \mu_p C_{ox} \frac{W}{L} (V_{gs} - V_{th})^2 (1 + \lambda V_{ds}) \quad (15)$$

The resulting tail current will decrease slightly because of the small variation in offset voltages, this minor variation will significantly change the g_m value, as shown in Eq. (16).

$$g_m = \sqrt{2 \mu C_{ox} \frac{W}{L} I_{ds}} \quad (16)$$

When we sweep trimming bits, the step size between two adjacent bits is one LSB different. From the analysis above, it can be deduced that the least significant bit (LSB) of g_m changes for every bit, so that the step size varies and the linearity of the CDS and CGS structure is affected by V_b variation. However, apart from this, the input pairs with source switches may also be affected by source degeneration effects because switches function as

linear resistors to degenerate input g_m , so the difference between CSS and BSS must be considered.

$$G_m = \frac{g_m}{1 + g_m R_{on}} \quad (17)$$

$$R_{on} = \frac{1}{\mu_p C_{OX} \frac{W}{L} (V_{gs} - V_{th})} \quad (18)$$

For example, assume for CSS that the g_m of LSB is G_{m1} , and since the trimming input pairs are binary-weighted, the transconductance of the second branch G_{m2} should be two times of G_{m1} , while

$$\frac{G_{m2}}{G_{m1}} = \frac{\frac{2g_{m1}}{1 + 2g_{m1}R_{on}}}{\frac{g_{m1}}{1 + g_{m1}R_{on}}} = 1 + \frac{1}{1 + 2g_{m1}R_{on}} < 2 \quad (19)$$

Since the actual g_m of second branch is therefore smaller than the desired value, non-linearity in step size is introduced, but it could be minimized by two methods.

The first method is to use BSS. Switches are sized in binary-weighted style to make the $(1 + g_m * R_{on})$ term constant and R_{on} is inversely proportional to the (W/L) ratio in Eq. (18).

$$\frac{G_{m2}}{G_{m1}} = \frac{\frac{2g_{m1}}{1 + 2g_{m1} * R_{on2}}}{\frac{g_{m1}}{1 + g_{m1} * 2R_{on2}}} = 2 \quad (20)$$

The second method would apply another structure called SSS. Instead of putting a switch at the source node, it is placed at the gate node and the input pairs are split into two transistors, as shown in Figure 4.1 (e). Typically, the (W/L) split ratio is easy to calculate using

$$\frac{W}{L_1 + L_2} = \frac{W}{L_1} + \frac{W}{L_2} \quad (21)$$

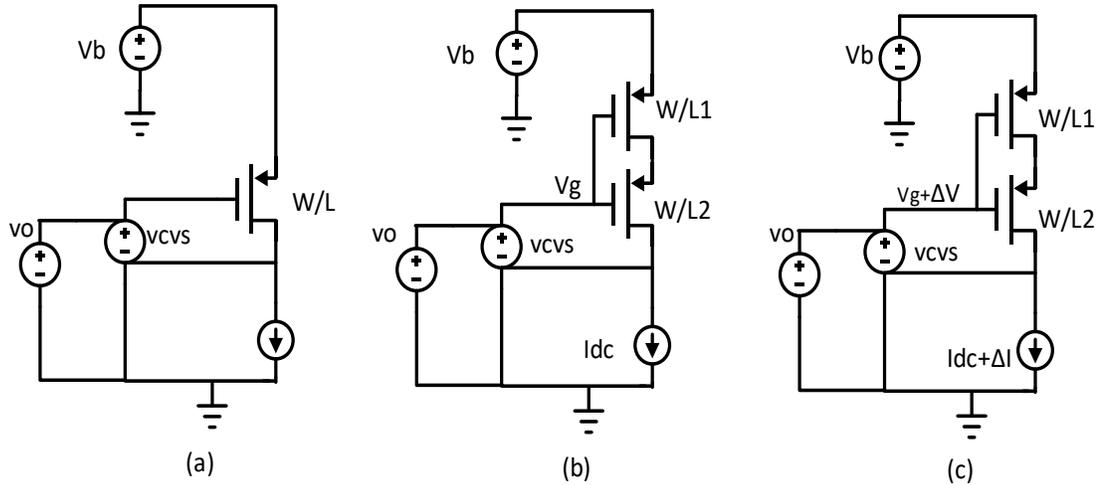


Figure 4.2 Testbench for g_m calculation (a) single transistor with size (W/L) , (b) split source, I_{dc} biased, (c) split source, $(I_{dc} + \Delta I)$ biased

But if V_{th} varies significantly with length, the equalized (W/L) ratio cannot be calculated with Eq. (21), and the testbench shown in Figure 4.2 can be used. $(L_1 + L_2)$ does not necessarily equal to L if

$$\Delta I / \Delta V = g_m \quad (22)$$

where g_m is the transconductance of the transistor in Figure 4.2(a).

Non-monotonicity issues can occur in the CSS structure. For example, consider what happens between trimming bit 15 (001111) and trimming bit 16 (010000). In that case, according to the switch control logic, the left side switches are all on and bit 16 opens most significant bit (MSB), while bit 15 opens all bits except MSB. Typically, bit 15 should generate larger offset because

$$g_{m1} + 2g_{m1} + 4g_{m1} + 8g_{m1} < 16g_{m1} \quad (23)$$

In other words, the g_m mismatch between the left side and the right side at bit 15 is larger than that at bit 16, while source degeneration makes this inequality also depends on R_{on} .

$$\frac{g_{m1}}{1+g_{m1}R_{on}} + \frac{2g_{m1}}{1+2g_{m1}R_{on}} + \frac{4g_{m1}}{1+4g_{m1}R_{on}} + \frac{8g_{m1}}{1+8g_{m1}R_{on}} > \frac{16g_{m1}}{1+16g_{m1}R_{on}} \quad (24)$$

$$g_{m1}R_{on} > 0.00666 \quad (25)$$

If Eq. (25) is valid, a non-monotonicity issue will arise, and although it is related to both g_m and R_{on} , we can optimize R_{on} to make it sufficiently small using transmission gate switch instead of single transistor switch. For equal PMOS and NMOS sizes, NMOS R_{on} is about one-fourth that R_{on} of PMOS due to different mobilities of charge carriers (for GlobalFoundry 0.13 μ m process), so the total parallel R_{on} of the transmission gate is much smaller than the R_{on} of a single PMOS.

However, non-monotonicity may not be an issue for both the linear search algorithm and the binary search algorithm. For the linear search algorithm, even though we might begin searching from least bit to largest bit or from largest bit to least bit, the trimming bit can always be found, showing existence of two available trimming bits. If, however, we can certify that the step size is small enough, this should be a valid approach. For the binary search this is also not a problem because the algorithm can always determine the best trimming code and make sure that the comparator offset after trimming is smaller than one step size. But, for the Newton's search, the chosen bit may generate an unwanted offset voltage if non-monotonicity occurs.

Integral nonlinearity (INL) and differential nonlinearity (DNL) define linearity quality for different schemes, as expressed in Eq. (26) and Eq. (27).

$$INL = \frac{\max\{V(i) - i * V_{LSB}\}}{LSB(ideal)}, \forall i = 0, 1, 2, \dots, 63 \quad (26)$$

$$DNL = \max\{INL(i+1) - INL(i)\} \quad (27)$$

If INL has a high enough quality after one offset measurement, the trimming bit can be calculated and applied directly (Newton's search), and the difference between the actual trimming curve and the ideal curve at each code should be acceptable. Large DNL similarly makes a resolution limitation. Once the trimming bit has been selected, the offset voltage after trimming should lie within that step range, and the quantization error, half of step size, should be tolerable.

4.1.2 Trimming Range

Trimming range (TR), determining total trimming bits that should be used, is usually estimated before design. If it is assumed that after-trimming offset is within one LSB voltage at a six-sigma range, then Monte Carlo (MC) simulation can be run to obtain one-sigma value. Six-sigma value is then the maximum-achievable offset voltage V_{max} . Trimming bit N can be expressed as

$$2^N * V_{LSB} \geq 2V_{max} \quad (28)$$

V_{max} must be multiplied by 2 because offset adds randomly to both input V_p and V_m . If TR is large enough, linearity is more critical than TR in achieving better resolution.

4.1.3 GBW

Pre-amplifier gain bandwidth product (GBW) determines how fast a comparator can make a decision, charge its output capacitor, and settle the output. Bandwidth (BW) is decided by first stage output impedance and loading capacitor, and all structures except CDS have the same BW because the input pairs are directly connected to the first stage

output. The BW of CDS may change a lot because its switch transistors are connected to the first stage output and total capacitance varies for different trimming bits. However, considering that the main input pairs are always on and their sizes are much larger than the combination of the trimming input pairs, this variation can be neglected. Table 4.1 shows that CDS and CGS produce similar values for GBW, implying that BW should be similar because these two structures have the same gain. For CSS, GBW becomes lower due to source degeneration, but the transmission gate switch with smaller R_{on} can help result in less g_m drop.

4.1.4 Area

A single transistor switch was not available for a source switch because of non-monotonicity problem and g_m degeneration issue. Although a transmission gate switch could tackle these two problems, the total area would be larger. For BSS, its performance is even better than CSS with a transmission gate, but more area is needed for the larger switch size.

4.2 Simulation Results

4.2.1 Trimming Curve

Table 4.1 summarizes properties of different switch scheme trimming curves, and Figure 4.3 shows that CGS and CDS have nearly the same good linearity. Effective trimming range means that only six sigma value offset is considered. In our case, six-sigma value is 12.48mV as calculated in last chapter. In order to compare INL and DNL of four types of switches, we should make them have similar effect trimming range. In Table 4.1, CSS always has the worst INL, which means they are not suitable for Newton's search, because the trimming resolution is very poor. The DNL, as defined, should be smaller than LSB, otherwise after-trimming offset voltage will be larger than target step size. According

to Table 4.1, CGS and CDS have the best DNL result, but CGS has the best INL result. In a word, CGS is the best switch structure to be applied. In Figure 4.3, different trimming curves are shown. As expected, CDS and CGS have the best linearity and resolution. Compared to CSS, BSS improves the linearity a lot.

Table 4.1 *Nominal case different trimming schemes parameters*

Scheme	INL	DNL	Effective TR (mV)	GBW(GHz)
CDS	1.67	0.97	12.78	6.289
CGS	1.59	0.96	12.78	6.217
SSS	1.70	0.98	12.63	6.167
CSS	2.54	0.99	12.53	5.461
BSS	1.57	0.94	12.83	5.963

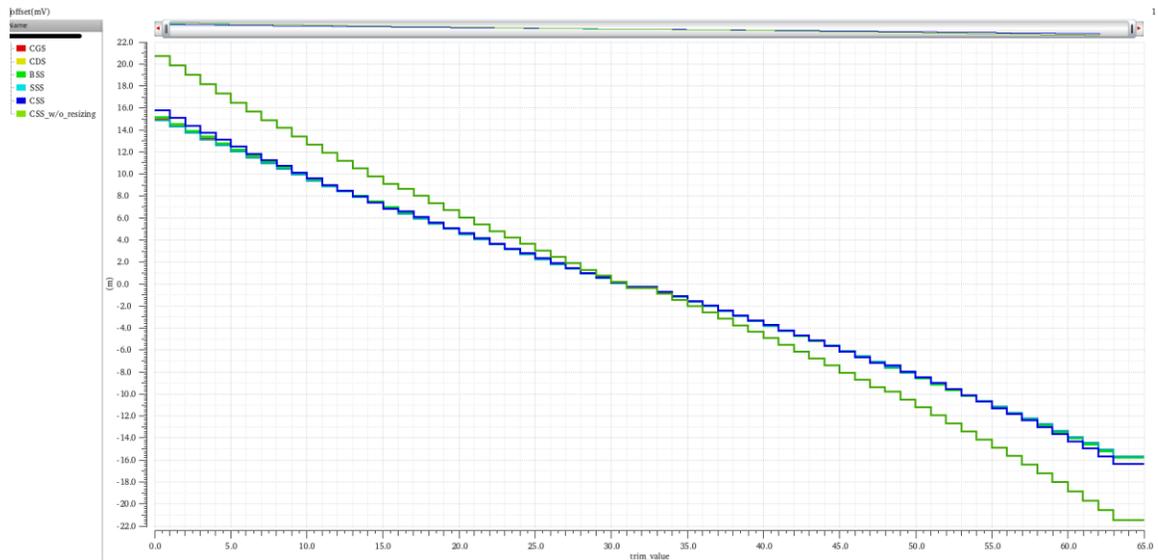


Figure 4.3 *Offset trimming curve comparison between five types of switches*

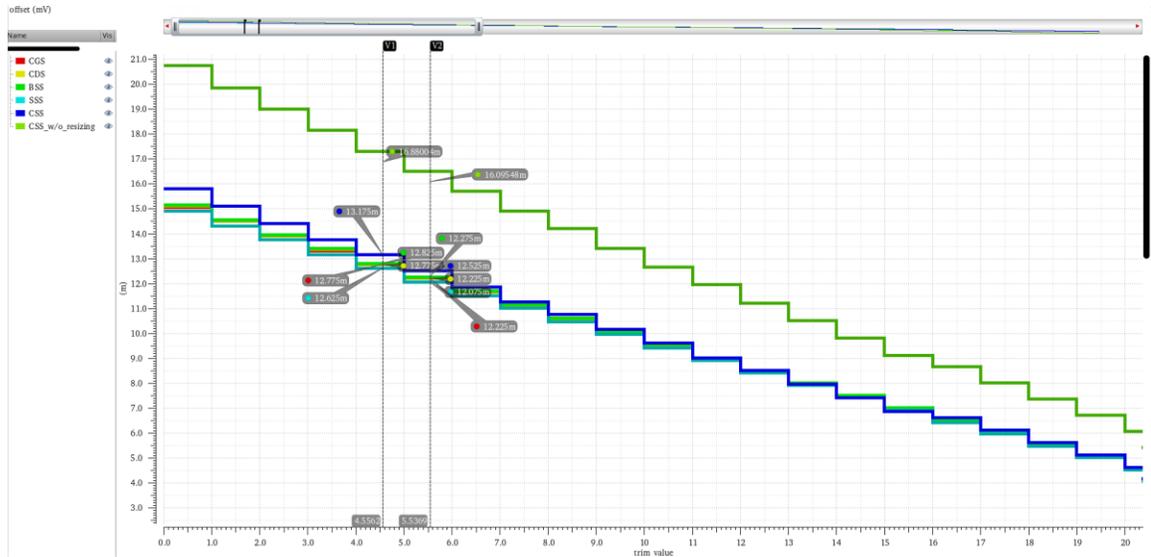


Figure 4.4 *Effective trimming range*

4.2.2 Post Trimming Offset with Binary Trimming Algorithm

We obtained post-trimming simulation results using the MATLAB trimming method introduced in preceding chapters. With 300 random offset values generated and totaled up at the input pairs, the binary search algorithm (see Appendix A) can start Cadence and obtain the after-trimming offset value for different input pair switch schemes.

In order to mimic real offset case, we can generate both process variations and random variations. Also, in order to show the trimming range limitation, we can purposely change the offset generated to a larger value. The histogram of offset we generate is shown in Figure 4.5. It shows that except for 3 points that are beyond trimming range, all others are distributed as a mix of normal distribution and uniform distribution. These 3 points are circled to make intentional calibration errors, and they should not be trimmable. Offset normal distribution is due to random variation, and offset uniform distribution is due to process variation.

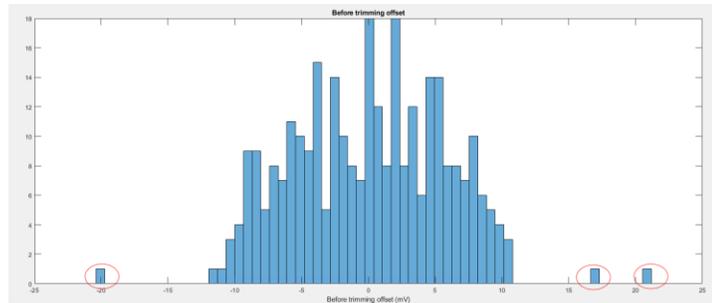
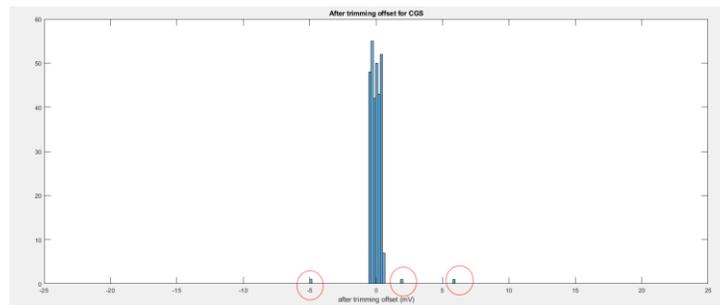
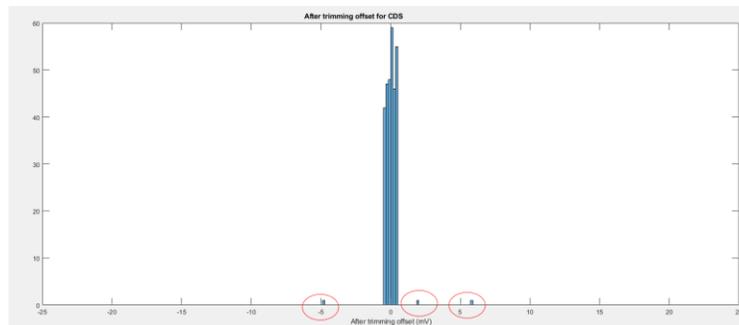


Figure 4.5 *Before trimming offset generation*



(a) CGS

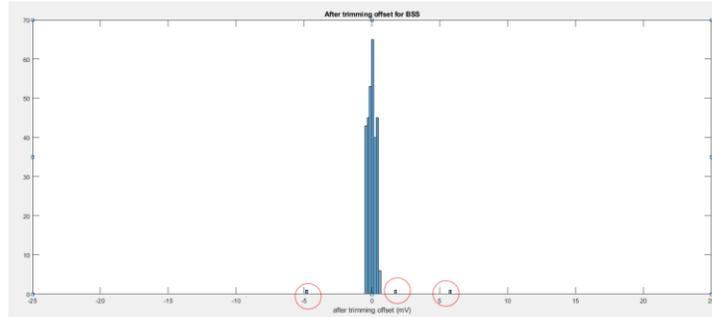


(b) CDS

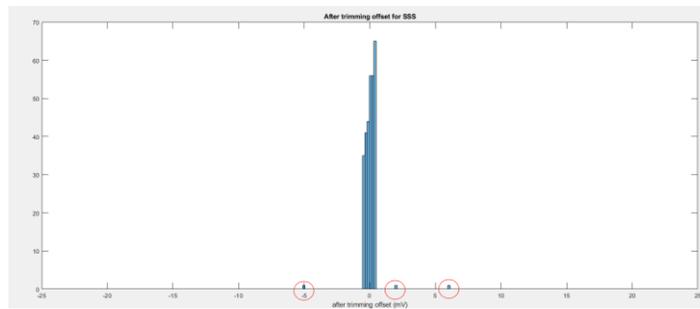
Figure 4.6 *After trimming offset for (a) CGS; (b) CDS; (c) BSS; (d) SSS; (e) CSS*

Figure 4.6 shows the after trimming offset results for five types of switches. Each of them has 3 points out of range due to intentional offset we added. We can see from the simulation histograms and the summarization table that CDS and CGS exhibit the best equivalent performance. For CSS, even though INL (shown in Table 4.1) is bad, the after

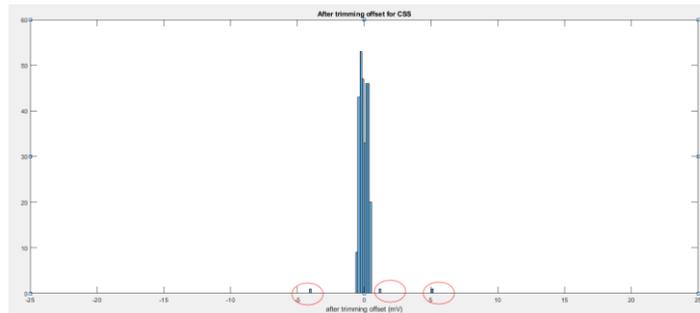
Figure 4.6 continued



(c) BSS



(d) SSS



(e) CSS

trimming offset target is met because DNL is less than 1. With almost same trimming range for all five types of switches, the offset trimming target is achieved. Notice that, CSS has one advantage: area is the smallest among five types of switches. This is because of nonlinearity issue. If same trimming transistors are used like other four types switches, the

trimming range of CSS will be bigger. CSS without resizing has about 20mV trimming range. In order not to over design trimming range, we can decrease CSS trimming transistors sizes. In this way, we will have similar trimming ranges for different switches, and it makes sense to compare linearity between them.

In order not to over design trimming range, we can decrease CSS trimming transistors sizes. In this way, we will have similar trimming ranges for different switches, and it makes sense to compare linearity between them.

In general, CGS is the easiest type of switch to be designed, and it exhibits the best performance when evaluated by both theoretical analysis and simulation results. The linearity of CGS is one of the best, and the after trimming offset target can be met. Table 4.2 shows the after trimming offset summary. The intended out of range offset is not included in Table 4.2. From results, all switches can achieve the trimming target.

Table 4.2 *After trimming offset voltage summary*

Maximum after trimming offset voltage (mV)	
Switch	Binary trimming
CDS	0.461
CGS	0.423
CSS	0.488
SSS	0.455
BSS	0.458

In Table 4.2, the offset voltage after applying the cancellation technique of this study is compared with previous comparator offset cancellation procedures. The after-trimming

offset value is determined by the target set and the trimming bits that can be implemented. In other previous works, different offset cancellation techniques are applied. In proposed design, $500\mu\text{V}$ offset can be achieved as the step size is smaller than $500\mu\text{V}$. In [22], A body-voltage trimming method is proposed as a high-resolution offset cancellation technique. The core idea is to use an analog feedback to change the body voltage so that the threshold voltage can be changed. In [23], a trimmable latch body voltage so that the threshold voltage can be changed. In [23], a trimmable latch comparator is presented, and the current flow through one branch of the comparator is trimmed by current mirrors ratios. Also, a feedback is necessary for detecting the offset

Table 4.3 *After-cancellation offset compared with other comparator circuits*

Item	This work	Ref [22]	Ref [23]	Ref [24]	Ref [25]
Technology	Analog $0.13\mu\text{m}$ CMOS	Analog $0.18\mu\text{m}$ CMOS	Analog $0.18\mu\text{m}$ CMOS	Analog CMOS	Analog $0.18\mu\text{m}$ CMOS
Offset	$500\mu\text{V}$	$750\mu\text{V}$	1.1mV	1mV	2.3mV
Power supply	3.3V	1.8V	0.5V	$\pm 5\text{V}$	1V
Power consumption	$49.5\mu\text{W}$	N/A	$28.5\mu\text{W}$	N/A	$90\mu\text{W}$

voltage. It can work under extremely low voltage and has the lowest power consumption. In [24], a trimming technique using floating gate and tunnel effect can be implemented to the entire system and can trim total offset. A parallel trimming method based on analogue

non-volatile memory is realized. In [25], Using similar body-voltage trimming idea, no extra trimming quiescent current is needed. Also, for a latched comparator, it can operate in a 5GHz high speed, but it consumes largest power, even under the lowest supply voltage.

CHAPTER 5. OP AMP OFFSET TRIMMING WITH BINARY SEARCH

5.1 Op Amp and trimming block structure

A simplified folded cascode amplifier with trimming blocks is shown below, and the trimming application should be different for operational amplifiers (Op amps) because input pairs cannot be trimmed. To achieve constant GBW and good stability, the input g_m should be constant, but the g_m will also vary if the input pair sizes are changing. Although trimming pairs cannot be implemented at the input stage, it can be implemented at the cascode stage.

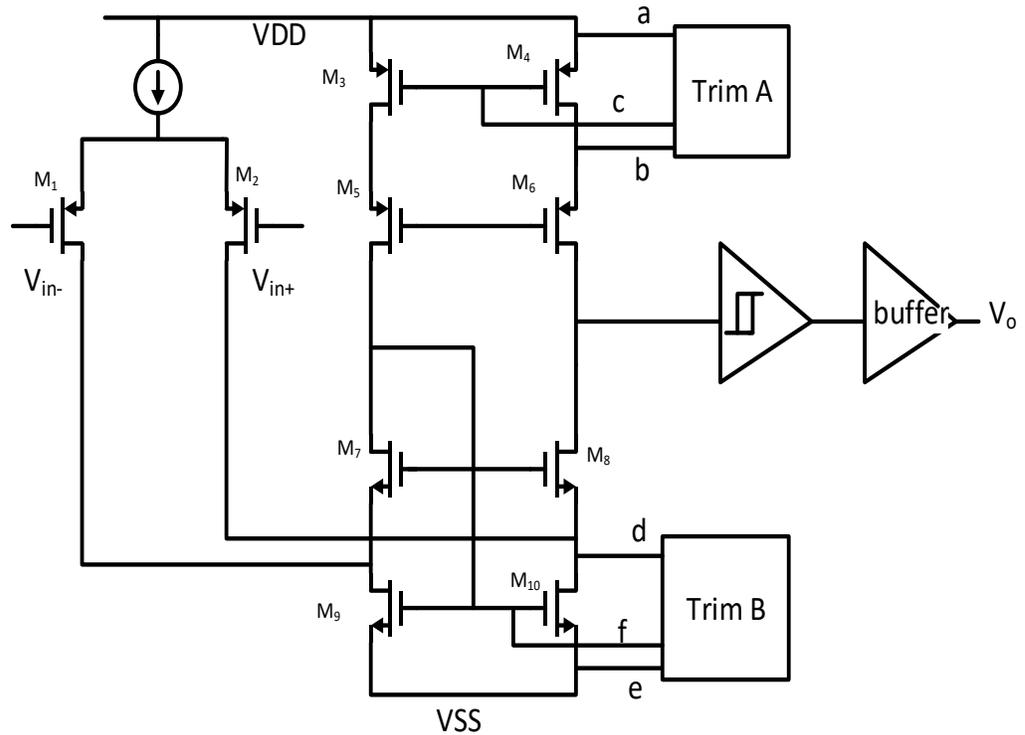
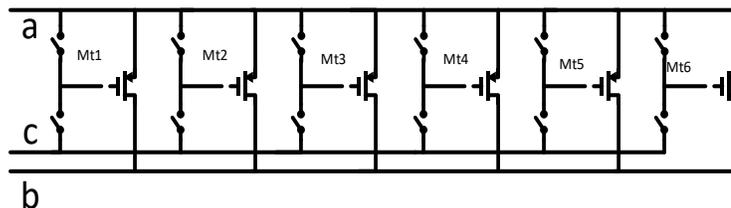


Figure 5.1 Op Amp structure



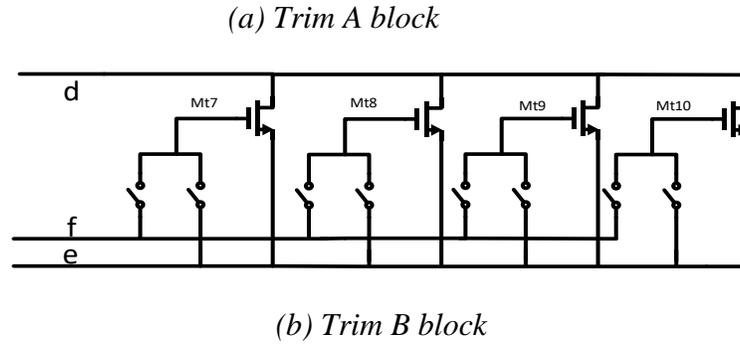


Figure 5.2 Op Amp with trimming blocks (a) Trim A (b) Trim B

5.2 Op Amp trimming analysis

Two trim blocks were used. Trim A is a fine trim including least significant bit (LSB) and Trim B is a coarse trim including most significant bit (MSB). A gate trimming switch is applied, and two switches controlled by two inverted signals can control one trimming branch. In Trim A, PMOS is used as switches, while in Trim B NMOS is used as switches. There is no need for a transmission gate, and the trimming transistors can be shorted to VDD/VSS or connected to the main pairs through gate connections. Transistor sizes in both Trim A and Trim B should be carefully selected so that the difference between offset generated by the smallest coarse trim bit and largest fine trim bit is smaller than one LSB. Otherwise, the entire range cannot be covered.

Trimming bits were calculated in same way, and Monte Carlo (MC) simulation is used to estimate the potential sigma. Our goal is to trim the offset of this Op Amp to be smaller than $25\mu\text{V}$ at a three-sigma value. Based on the simulation, the sigma value is 2.51mV , i.e., three-sigma value would be 7.53mV . This means that the total number of decimal values required for covering half of total trimming range would be $7.53/0.025 = 301.2$. So, considering that the offset can be either positive or negative, ten binary bits are needed, and each side would have 512 trimming decimal bits.

In the comparator offset trimming scheme, the offset is directly added to input pairs so we can easily recognize that the offset generated is proportional to transistor size. However, if trimming blocks are not directly implemented on input pairs, the linearity of the trimming decimal bits versus offset voltage should be examined, because we always want the trimming curve to be as linear as possible to support adequate estimation of the trimming range and the smallest step size. However, if we apply the trimming blocks to cascode stages, the linearity is still good enough, as proved by the following proof:

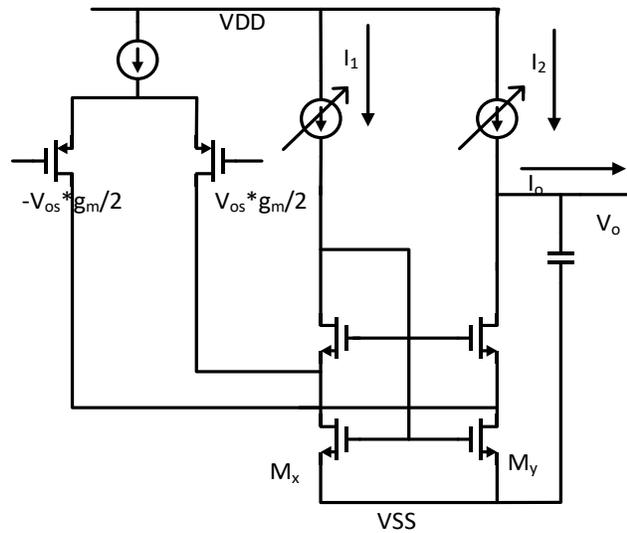


Figure 5.3 Op Amp trimming on cascode stage

Suppose that

$$I_1 : I_2 = 1 : (1 + m) \quad (29)$$

Then

$$I_{M_x} = I_1 + \frac{V_{os}}{2} * g_m = I_{M_y} = I_2 - I_0 + \frac{-V_{os}}{2} * g_m \quad (30)$$

Set

$$I_0 = 0 = I_2 - I_1 - V_{os} * g_m = m * I_1 - V_{os} * g_m \quad (31)$$

$$V_{os} = m * \frac{I_1}{g_m} \quad (32)$$

Similarly, suppose that

$$I_1 : I_2 = (1 + m) : 1$$

$$I_{My} = I_2 - I_o - \frac{V_{os}}{2} * g_m = I_{MX} = (1 + m)I_2 + \frac{V_{os}}{2} * g_m \quad (33)$$

Set

$$I_0 = 0 = -m * I_2 - V_{os} * g_m$$

$$V_{os} = m * \frac{I_2}{g_m} \quad (34)$$

In these derivations, m is the transistor size that can be controlled by trimming bits, and it is proportional to the (W/L) ratio of the transistors. From the equations, we can see that trimming on both sides should have good linearity between the offset value and the decimal trimming bits.

Similarly, the trimming transistors are binary-weighted sized, and 1024 is the total decimal trimming-bit range. At the middle point, 512 should ideally represent zero offset. However, instead of adding trimming branches on both sides as we did for the comparator, and it is also possible to equip only one side with a trimming adjustment. Even though the offset can be either positive or negative, it is still trimmable with one-sided trimming method. In Figure 5.1, we set M_9 and M_{10} , as well as M_3 and M_4 , to initially be intentionally asymmetrical. After adding the MSB of each trim block, the offset generated by asymmetry can be cancelled, meaning that the offset voltages generated equal, namely, $M_4 + Mt_6 = M_3$, $M_{10} + Mt_{10} = M_9$. In this way, binary trimming bit 1000100000 (544), rather than 1000000000 (512), should be the “middle” point where no offset exists. Binary trimming bit 0000000000 (0) will generate the largest negative offset and binary trimming bit

1111111111 (1023) will generate the largest positive offset. The only disadvantage of one-sided trimming is that the trimming range will be unsymmetrical because the middle point is not 512, but this may not be a problem if the trimming range is large enough to cover three-sigma offset voltages. In other words, if the missing range is not used, the three-sigma offset is still trimmable.

5.3 Transistor size summary

In Table 5.1. We can see that M3 and M4, or M9 and M10, do not have the same size. Transistors in the trimming blocks are binary-weighted.

Table 5.1 *Op Amp transistor size*

Transistor number	W/L (um)	Transistor number	W/L (um)
M ₁	(12/1) *6	M _{t1}	0.16/64
M ₂	(12/1) *6	M _{t2}	0.16/32
M ₃	4/1	M _{t3}	0.16/16
M ₄	0.85/4	M _{t4}	0.16/8
M ₅	2/0.8	M _{t5}	0.16/4
M ₆	2/0.8	M _{t6}	0.16/2
M ₇	4/0.6	M _{t7}	0.16/38
M ₈	4/0.6	M _{t8}	0.16/19
M ₉	1/3	M _{t9}	0.16/9.5
M ₁₀	0.8/3	M _{t10}	0.16/4.75

5.4 Simulation results

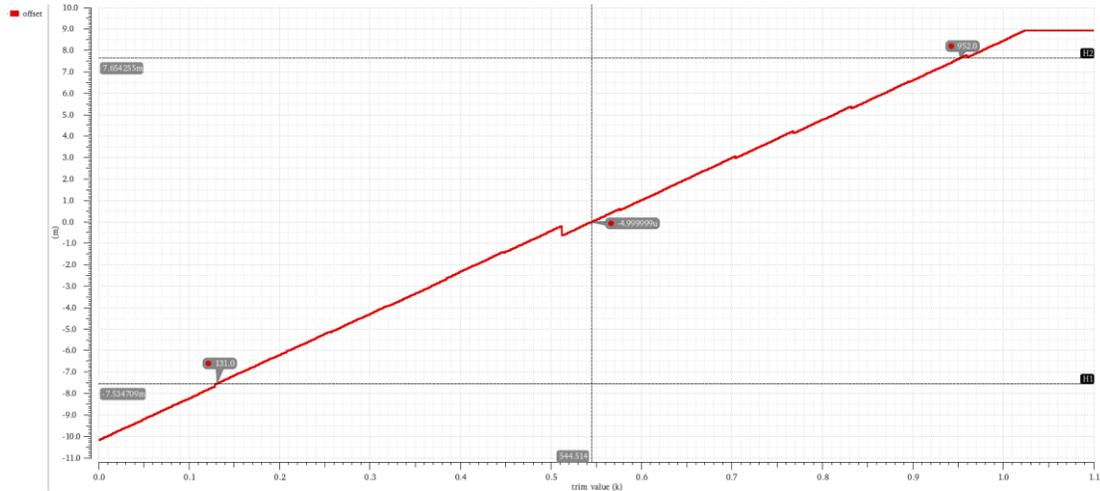
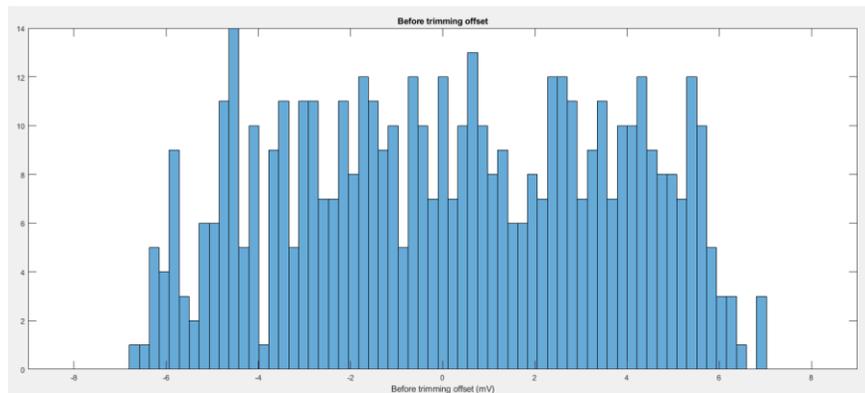


Figure 5.4 *Op Amp trimming range vs. trimming bits*

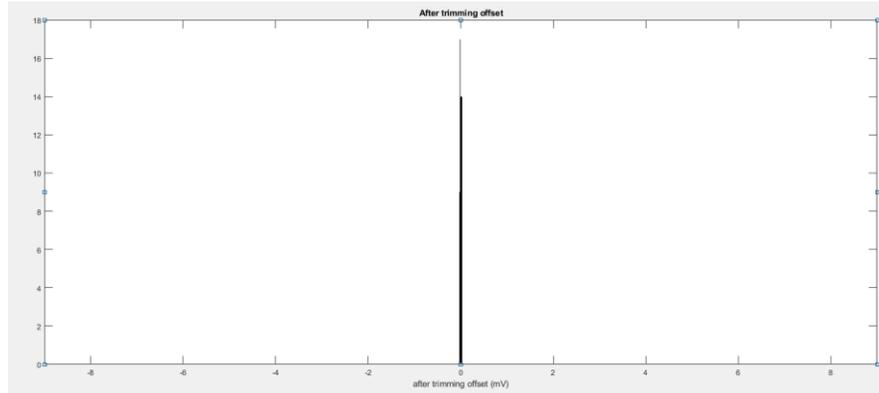
Figure 5.4 gives us a plot of offset vs trimming bits, showing that the trimming range is sufficiently large to cover 7.53mV, and the step size is less than 25 μ V, as expected. Non-monotonicity is not an issue with the binary search method either, because the trimming method can always find a correct solution. Using the same MATLAB method, we can obtain the histograms shown in Figure 5.5 for both before-trimming offset and after-trimming offset. Results show that the maximum after-trimming offset is 23 μ V after 500 runs, which is the desired target.



(a) offset before trimming

Figure 5.5 *Op Amp before and after-trimming offset voltage histogram*

Figure 5.5 continued



(b) offset after trimming

CHAPTER 6. CONCLUSION

This paper has introduced a comparator that includes a self-trimming algorithm. The auto-calibration system can be widely used to achieve precise small offset target. For the comparator, the design is implemented on GlobalFoundry 0.13 μ m process, and the design details are analyzed, focusing on offset voltage. Trimming is applied with MATLAB and Cadence, and trimming methods are analyzed and explained. Binary search algorithm is the core part of trimming digital blocks. Detailed coding is listed in Appendix A and B. The MATLAB method is implemented for five distinct types of trimming branch switches. The characteristics of each one is analyzed, and simulation results are evaluated and compared. From these results, we determine that a constant size gate switch (CGS) is the optimal choice for trimming switch control.

Then, using a similar approach, operational amplifier (Op Amp) trimming is evaluated using MATLAB to achieve the desired target. The differences between trimming a comparator and an op amp are pointed out with analysis and transistors level application.

REFERENCES

- [1] T. C. Carusone, D. A. Johns, and K. W. Martin, *Analog Integrated Circuit Design*, 2nd ed., Wiley, 2010, pp.413-418.
- [2] Y. Chiu, B. Nikolic', and P. R. Gray, "Scaling of analog-to-digital converters into ultra-deep-submicron CMOS," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2005, pp. 376–378.
- [3] C. Wu and J. Yuan, "An 11b 450MS/s three-way time-interleaved subranging pipelined-SAR ADC in 65nm CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 5, 2016, pp. 1223-1234.
- [4] C. Chiang and C. Chen, "Zero-voltage-switching control for a PWM buck converter under DCM/CCM boundary," *IEEE Transactions on Power Electronics*, vol. 24, no. 9, 2009, pp. 2120-2126.
- [5] G. C. Temes and C. Enz. "Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization," *Proceedings of the IEEE*, vol. 84, no. 11, 1996, pp. 1584-1614.
- [6] D. R. Holberg and P. E. Allen, *CMOS Analog Circuit Design*, 3rd edition, Oxford University Press, 2011, pp. 443-447.
- [7] R. Burt and J. Zhang, "A micropower chopper-stabilized operational amplifier using a SC Notch Filter With Synchronous Integration Inside the Continuous-Time Signal Path," *IEEE J. Solid-State Circuits*, vol. 41, no. 12, 2006, pp. 2729-2736.
- [8] R. F. Bullag, R. C. Ortega, and S. B. Bullag, "Adaptive trimming test approach-the efficient way on trimming analog trimmed devices at wafer sort," *36th International Electronics Manufacturing Technology Conference*, 2014, pp. 1-4.
- [9] D. J. Allstot, "A precision variable-supply CMOS comparator," *IEEE J. Solid-State Circuits*, vol. 17, no. 6, 1982, pp. 1080-1087.
- [10] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 2nd edition, Mc Graw Hill, 2017, pp. 603-604.
- [11] M. Bolatkale, A. P. Pertijs, J. Kindt, H. Huijsing, and A. A. Makinwa, "A single-temperature trimming technique for MOS-input operational amplifiers achieving 0.33 μ V/ $^{\circ}$ C offset drift," *IEEE J. Solid-State Circuits*, vol. 46, no. 9, 2011, pp. 2099-2107.

- [12] S. J. Lovett, M. Welten, A. Mathewson, and B. Mason, "Optimizing MOS transistor mismatch," *IEEE J. Solid-State Circuits*, vol. 33, no. 1, 1998, pp. 147–150.
- [13] P. R. Kinget, "Device mismatch and tradeoffs in the design of analog circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 6, 2005, pp. 1212–1224.
- [14] P. G. Drennan and C. C. McAndrew, "Understanding MOSFET mismatch for analog design," *IEEE J. Solid-State Circuits*, vol. 38, no. 3, 2003, pp. 450–456.
- [15] H. M. Staudt, "Comparator based self-trim and self-test scheme for arbitrary analogue on-chip values," *IEEE IMS3TW*, 2010, pp. 1-6.
- [16] Y. Zhu, C. Chan, U. Chio, S. Sin, F. Maloberti et al., "A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 6, 2010, pp. 1111-1121.
- [17] J. He, S. Zhan, D. Chen, and R. L. Geiger, "Analyses of Static and Dynamic Random Offset Voltages in Dynamic Comparators," *IEEE Transactions on Circuits and System I: Regular Papers*, vol. 56, no. 5, pp. 911–919, May 2009.
- [18] M. J. Pelgrom, A. C. Duinmaijer, and A. P. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989.
- [19] X. Wang "A low offset dynamic comparator with morphing amplifier," *Graduate Theses and Dissertations*, 2017, 16237.
- [20] N. Danniell. "SemiWiki - All Things Semiconductor." *SemiWikicom* RSS, 2011, www.semiwiki.com/forum/f119/semiconductor-process-variation-wiki-443.html.
- [21] W. Sansen, *Analog Design Essentials*, 1st edition, Springer, 2006, pp. 421-425.
- [22] S. B. Mashhadi and R. Lotfi, "An Offset Cancellation Technique for Comparators Using Body-voltage Trimming," *IEEE 9th International New Circuits and systems conference*, 2011, pp. 273-276.
- [23] M. Mohammadi and D. Sadeghipour, "A 0.5V 200MHz Offset Trimmable Latch Comparator in Standard 0.18um CMOS Process," *Iranian Conference on Electrical Engineering (ICEE)*, 2013, pp. 1-4.
- [24] M. Zhang , F. Devos, J.-F. Pone and Y. Ni, "A Parallel Trimming Method of Offset Reduction for Comparators and Amplifiers," *Proceedings of IEEE International Symposium on Circuits and Systems*, vol.5, 1994, pp. 715-718.

- [25] Y. Xu, L. Belostotski and J.W. Haslett, "Offset-Corrected 5GHz CMOS Dynamic Comparator using Bulk Voltage Trimming: Design and Analysis," *IEEE 9th International New Circuits and systems conference*, 2011, pp. 277-280.

APPENDIX A. BINARY SEARCH MATLAB CODE

```
vos_p = rand ([1,300]) * 0.019 - 0.0095;
vos_r = randn ([1,30]) * 0.001;
vos = vos_r + vos_p;
for i = 1:300;
    vos(i) = vos_g (1,i);
    trim_value(i) = 32;
    cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'
    cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',
string(vos(i)),');
    cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/bin_2.ocn;')
    dos(cmd);
    filename = '\\h10.ece.iastate.edu\xygong\bin_2_result';
    fid = fopen(filename,'r');
    A = fscanf(fid,'%e',[1 inf]);
    t(i) = A(:,1);
    fclose(fid);
    if t(i) > 1
        compout(i) = 1;
    else
        compout(i) = 0;
    end
    if (compout(i) == 1);
```

```

left_bit(i) = 0;

right_bit(i) = 31;

p(i) = 1; %polarity

else

left_bit(i) = 32;

right_bit(i) = 63;

p(i) = 0;

end

trim_value(i) = floor((left_bit(i) + right_bit(i))/2);

T(i,1) = trim_value(i);

cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'

cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',

string(vos(i)),');

cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/bin_2.ocn;') %

measure high or low

dos(cmd);

filename = '\\h10.ece.iastate.edu\xygong\bin_2_result';

fid = fopen(filename,'r');

B = fscanf(fid,'%e',[1 inf]);

u(i) = B(:,1);

fclose(fid);

if u(i) > 1 % t, u,v ... = 3.3 or 0

compout(i) = 1;

```

```
else
    compout(i) = 0;
end
if (compout(i) == 1);
    right_bit(i) = trim_value(i) - 1;
else
    left_bit(i) = trim_value(i) + 1;
end
trim_value(i) = floor((left_bit(i) + right_bit(i))/2);
T(i,2) = trim_value(i);
cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'
cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',
string(vos(i)),');
cmd=strcat(cmd, ' ocean -nograp -restore ~/ibm130/cmrf8sf/bin_2.ocn;')
dos(cmd);
filename = '\\h10.ece.iastate.edu\xygong\bin_2_result';
fid = fopen(filename,'r');
C = fscanf(fid,'%e',[1 inf]);
v(i) = C(:,1);
fclose(fid);
if v(i) > 1
    compout(i) = 1;
else
```

```
    compout(i) = 0;
end
if (compout(i) == 1);
    right_bit(i) = trim_value(i) - 1;
else
    left_bit(i) = trim_value(i) + 1;
end
trim_value(i) = floor((left_bit(i) + right_bit(i))/2)
T(i,3) = trim_value(i);
cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'
cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',
string(vos(i)),');
cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/bin_2.ocn;')
dos(cmd);
filename = '\\h10.ece.iastate.edu\xygong\bin_2_result';
fid = fopen(filename,'r');
D = fscanf(fid,'%e',[1 inf]);
w(i) = D(:,1);
fclose(fid);
if w(i) > 1
    compout(i) = 1;
else
    compout(i) = 0;
```

```
end

if (compout(i) == 1);

    right_bit(i) = trim_value(i) - 1;

else

    left_bit(i) = trim_value(i) + 1;

end

trim_value(i) = floor((left_bit(i) + right_bit(i))/2)

T(i,4) = trim_value(i);

cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'

cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',

string(vos(i)),');

cmd=strcat(cmd, ' ocean -nograp h -restore ~/ibm130/cmrf8sf/bin_2.ocn;')

dos(cmd);

filename = '\\h10.ece.iastate.edu\xygong\bin_2_result';

fid = fopen(filename,'r');

E = fscanf(fid,'%e',[1 inf]);

x(i) = E(:,1);

fclose(fid);

if x(i) > 1

    compout(i) = 1;

else

    compout(i) = 0;

end
```

```

if (compout(i) == 1);
    right_bit(i) = trim_value(i) - 1;
else
    left_bit(i) = trim_value(i) + 1;
end
trim_value(i) = floor((left_bit(i) + right_bit(i))/2);
T(i,5) = trim_value(i);
if (p(i) == 1)
    final_trim(i) = trim_value(i) + 1;
else
    final_trim(i) = trim_value(i);
end
T(i,6) = final_trim(i);
cmd='c:\Program Files\PuTTY\plink -l xygong research-8.ece.iastate.edu'
cmd=strcat(cmd, ' export trim_value=', string(final_trim(i)), ' vos=',
string(vos(i)),');
cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/lala.ocn;')
dos(cmd);
filename = '\\h10.ece.iastate.edu\xygong\lala_offset';
fid = fopen(filename,'r');
F = fscanf(fid,'%e',[1 inf]);
aftertrim_offset(i) = F(:,1);
fclose(fid); end

```

APPENDIX B. NEWTON'S SEARCH MATLAB CODE

```

vos_p = rand ([1,300]) * 0.019 - 0.0095;

vos_r = randn ([1,30]) * 0.001;

vos = vos_r + vos_p;

range = 12.5e-03 % trim range

ave = range / 27 % step

for i = 1:50;

    vos(i) = vos_g (1, i);

    if vos(i) >= 0

        trim_value(i) = round (31 - vos(i) / ave);

    else

        trim_value(i) = round (32 + abs(vos(i)) / ave);

    end

    cmd='c:\Program Files"\PuTTY\plink -l xygong research-8.ece.iastate.edu'

    cmd=strcat(cmd, ' export trim_value=', string(trim_value(i)), ' vos=',

string(vos(i)),');

    cmd=strcat(cmd, ' ocean -nograph -restore ~/ibm130/cmrf8sf/lala.ocn;')

    dos(cmd);

    filename = '\\h10.ece.iastate.edu\xygong\lala_offset';

    fid = fopen(filename,'r');

    A = fscanf(fid,'%e',[1 inf]);

    t(i) = A(:,1);

    fclose(fid); end

```