AN ABSTRACT OF THE THESIS OF

Sachin Ranganathan for the degree of Master of Science in

Electrical and Computer Engineering presented on April 15, 2003.

Title:

Design of a Variable Gain, High Linearity, Low Power Baseband Filter for WLAN Transmitters

Redacted for privacy

Abstract approved: _____

Terri S. Fiez

A variable gain, high linearity, low power baseband filter for WLAN applications is implemented in a 1.5 V 3 V 0.15 μm CMOS process. This fourth-order low-pass filter, which is introduced in the transmit channel as a reconstruction filter between the D/A converter and the mixer, has a measured cut-off frequency of 9 MHz. The active-RC configuration has single amplifier biquads (SABs) to save power and is implemented using three-stage opamps with nested-Miller compensation for better linearity. It also features a special π-to-Tee transformation network for the resistor arrays, used for frequency or gain trimming, in order to obtain higher linearity than conventional Sallen-Key circuits. The measured THD for a 2 V_{p-p} signal at 1 MHz is -72 dB.

Design of a Variable Gain, High Linearity, Low Power Baseband Filter for WLAN Transmitters

by

Sachin Ranganathan

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed April 15, 2003
Commencement June 2004

Master of Science thesis of Sachin Ranganathan presented on April 15, 2003

APPROVED:

Redacted for privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for privacy

Director of the School of Electrical Engineering and Computer Science

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Sachin Ranganathan, Author

ACKNOWLEDGMENT

First and foremost, I would like to express my profound gratitude in thanking my advisor, Dr. Terri Fiez. She has been a constant source of support during every stage of my research and has provided excellent guidance as a mentor. She has also helped in the improvement of my presentation and communication skills with her insight and observations in the details of my work. I would also like to thank her for the freedom that she gave me which enabled me to work independently and produce quality results.

Secondly, I would specially like to thank Prakash Easwaran and Rupak Ghayal from Texas Instruments for all their technical help and ideas that they provided towards my research. They were always available for assistance and I felt that my skills were immensely honed by their constructive reviews. I learned a lot about system and circuit design through their experience and it was a pleasure to work with them.

I am immensely grateful to Gani Subramaniam, my supervisor at Texas Instruments (TI), and Brian Evans for giving me an opportunity to be part of their group which has helped in defining my thesis. I thank Charolette Garza and Walt at TI for their help with layout issues. I would also like to thank C.Srinivasan, Christian Yots and Francesco for their assistance at TI and in helping me to meet my deadlines. I also appreciate the assistance provided by Ali Dweik, Joe Venable, Mark Eck, Ken Hudson, Mark Rodgers and Brandon Beckham in doing my measurements at TI. I thank Texas Instruments for undertaking the fabrication of my chip and making the completion of my thesis possible.

I would like to express my appreciation for Dr. Karti Mayaram for being instrumental in my coming to the United States and giving me the liberty to choose an appropriate advisor for my research. I thank Dr. Gabor Temes and Dr. Un-Ku Moon for the courses that they taught me which helped in developing skills that I could use for my research. I am thankful to Ferne Simendinger, April Melton and Sarah O' Leary for all the help provided during my academic stay at OSU.

I would like to take this opportunity to express my deepest gratitude to Jose Silva who has taken time out of his own research to provide selfless support and assistance with the installation of related software tools from Texas Instruments at OSU. I also thank Richard White at TI who helped out with the same.

Being a part of the AMS Lab has been really encouraging in providing an appropriate work atmosphere for a student to thrive on. I would like to thank Robert, James, Paul, Scott, Zhimin, Volodymyr, Taras, Chenggang, Yutao, Shengjun, Kalyan, Vinay, Ravikanth, Ajit and Patrick for the help they have provided in the lab. I am also thankful to Prashant, Anurag Tiwari, Anurag Pulincherry, Thirumalai, Brian, Arun, Kala, Prachi, Kannan, Krishnaprasad, Manas, Manu, Desikachari, Raghuram, Nitin, Kaustubh, Sanjay, Namita, Omkar, Eashwar, Triet, Shu-Ching, Husni, Mohana, Gautam, Sirisha, Vivek, Kingsuk, Supriya, Josh, Ravi, Priya, Avantika, Monika, Dipesh, Annika, Maithili, Sunita, Madhuri, Reshma, Aarti, Shruti, Roshan, Ragini, Ramya, Meghana, Apeksha and Luke for making my days at Corvallis memorable.

As for my friends at Texas Instruments, I would like to thank Aline, Shanthi, Madhu and Neel for making my stay in Dallas enjoyable. I am indebted to Neel who has supported me in Corvallis and Dallas and made my stay comfortable while

working at Texas Instruments. I am also grateful to Madhu for helping me in my research at Texas Instruments.

I would like to thank my alma maters, Campion, D.G.Ruparel, B.I.T.S. Pilani and Oregon State University for providing me with the education I needed and helping me become what I am as I step out into the professional world. I take this chance to express my heartfelt gratitude to Ishmeet, Sumeet and Manoj for being supportive of me through my stay in India and the United States and whose friendships I have always cherished.

Last, but not least, I am deeply indebted to my parents and my sister, Sweta, who have been with me through every phase of my life and whose sacrifices and love have brought me where I am today. I thank them for instilling confidence in me and trusting me during my ups and downs in life and shall be grateful to them forever. I am also very grateful to my grandparents and relatives who have supported me in everything that I have done. I thank God for everything.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

Figure | Page

LIST OF TABLES

DESIGN OF A VARIABLE GAIN, HIGH LINEARITY, LOW POWER BASEBAND FILTER FOR WLAN TRANSMITTERS

1. INTRODUCTION

1.1. Motivation

The increasing volume of personal communication devices has resulted in various services and standards of communication such as GSM, DECT, Bluetooth and 802.11. Wireless local area networks (WLANs) are rapidly emerging as one of the leading technologies that can support high data rates in offices and other work places. WLAN applications include laptop computers and other portable items whose consumers are more interested in low cost, small area and low power than high performance. To meet these constraints, it is desirable to realize these devices as a single integrated circuit. A major challenge in doing this is integrating the analog circuitry using the same technology as that used for the digital circuit implementation. The power dissipated by digital circuits is proportional to the square of the supply voltage and therefore low supply voltages are preferred to reduce the power consumption [1]. This has led to technology scaling in CMOS processes where the voltage is scaled down based on electric field restrictions. Hence, to incorporate the complete system on a single chip and have a single supply voltage to reduce cost, it is advantageous to operate the analog circuitry off of these low supply voltages. Thus, transmitter and receiver designs to incorporate the whole system on a chip for these standards are becoming increasingly challenging considering the various trade-offs involved in terms of power, area, noise, linearity and other factors [2] - [9].

An important component of the transmitter is the baseband continuous-time filter that smoothes the output of the D/A converter before up-conversion by the

mixer as depicted in Fig. 2.1. This filter must suppress all images from the D/A converter. It should have an in-band signal-to-noise ratio greater than 40 dB, an out-of-band noise floor below 30 nV/$\sqrt{\text{Hz}}$ and distortion levels below -50 dB. Intermodulation distortion is another serious problem in wide-band applications with multi-tone signals. Two tones introduced in the signal band can produce other unwanted tones within the signal band. These must be suppressed to a considerable degree as compared to the original tones. Hence, the need to make the filter highly linear with low operating supply voltages while also maintaining low power consumption increases the challenges for the baseband filter design. In addition, incorporating variable gain into the filter eliminates the need for an additional variable gain amplifier after the smoothing filter to adjust varying signal amplitude levels.

1.2. Research Goals

There are various kinds of WLAN standards, each having its own set of specifications, some of which are Bluetooth, 802.11a, 802.11b, etc. Each standard may have a different modulation and carrier scheme for its signal which determines the effective bit rate. This thesis focuses on the design of a high linearity, low power, low area continuous-time baseband filter for a WLAN 802.11b transmitter. It has been designed for a 11 MHz signal bandwidth and supports QPSK (Quadrature Phase-Shift Keying) and OFDM (Orthogonal Frequency Division Multiplexing) signals. The filter is designed in a 0.15 μm CMOS process and for a 1.5 V power supply. The high linearity and low power to incorporate the variable gain is accomplished using an active-RC configuration with single amplifier biquads and with an additional frequency and gain trimming scheme to improve the distortion performance.

1.3. Thesis Organization

Chapter 2 presents a typical transmitter architecture and emphasizes the role of the baseband filter in the transmitter. The specifications for the filter are derived from the specifications of the whole system and other blocks around the filter. Chapter 3 describes the design and implementation of the baseband filter and elaborates on the filter architecture, the operational amplifier circuit and the hybrid trimming circuit used. The measurement results and other layout issues are discussed in Chapter 4 followed by the conclusions from this work in Chapter 5.

2. SYSTEM ARCHITECTURE AND SPECIFICATIONS

2.1. Transmitter Architecture

Transmitters are a key part of communication transceivers. The exact architecture of the transmitter can vary depending on the particular application and the trade-offs involved. The basic function of a transmitter is conversion of the digital signal to analog, filtering of this signal to reconstruct the baseband signal and suppress image signals from the D/A converter and up-conversion to the RF. This may be followed by some RF filtering and power amplification to achieve the final output power level. A number of transmitter architectures exist which cater to the various communication applications and system requirements of the particular application.

2.1.1. Direct-Conversion Transmitter

A basic and simple architecture is that of a direct-conversion transmitter shown in Fig. 2.1, where I and Q denote the in-phase and quadrature-phase signals, respectively [6]. In this transmitter, the digital signal is converted to an analog signal by a D/A converter and then the mixer modulates and up-converts the signal to the carrier radio frequency (RF) in a single step. This may be followed by a band-pass filter to suppress harmonics of the local oscillator (LO) and a power amplifier (PA) to boost up the signal strength before transmitting the signal. The PA output may cause some disturbance to the local oscillator as it is a high power signal centered around the LO frequency and hence may corrupt the LO output. Although this

architecture is somewhat sensitive, it is simple and very convenient for integration on a single chip.

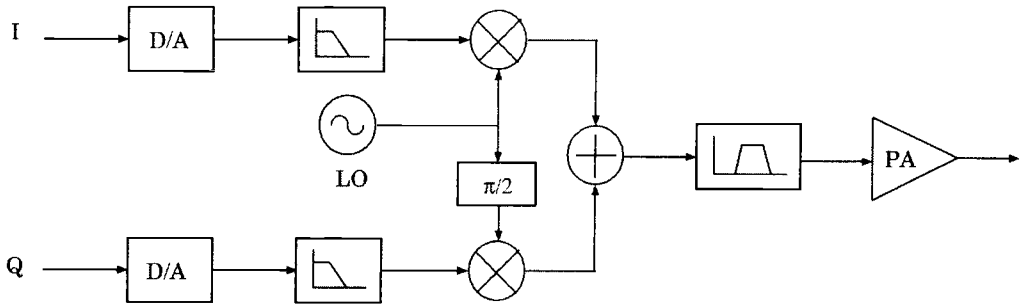


Figure 2.1. Direct-conversion transmitter.

2.1.2. Heterodyne Transmitter

Heterodyne transmitters [7] differ from direct-conversion transmitters in that the transition from baseband to RF is done in two steps; the baseband signal after the D/A converter undergoes quadrature modulation to an intermediate frequency (IF) by the first mixer and then is up-converted to RF with another mixer which has a different LO frequency from the first mixer. This avoids the PA disturbance issue which exists in the direct-conversion architecture since the PA center frequency is far away from either of the LO frequencies. Fig. 2.2 shows the block diagram of a typical heterodyne architecture.

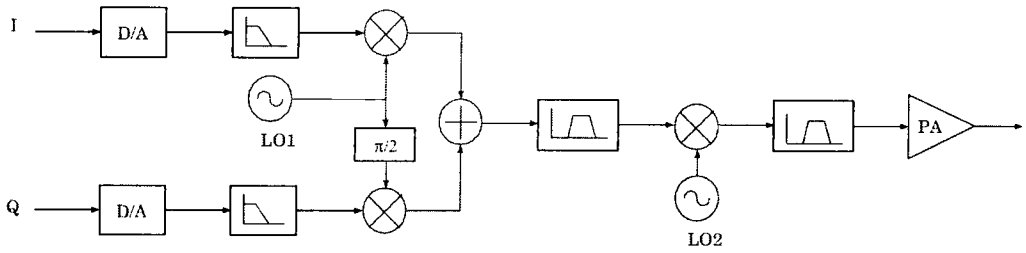


Figure 2.2. Heterodyne transmitter.

The disadvantage of this architecture is that it requires a filter, centered at RF, after the second mixer to reject the sideband produced during this up-conversion. Moreover, this architecture is more complicated and consumes more power and chip area than the homodyne architecture.

2.1.3. PLL-based Transmitter

A slightly different technique is to use a phase locked loop (PLL) in the transmitter as shown in Fig. 2.3 [8]. The frequency translation occurs in two steps, like the heterodyne architecture, but in this approach, the IF is converted to RF with a PLL instead of a simple mixer. In the PLL, the mixer down-converts the RF to IF which then serves as the reference frequency for the phase detector (PD) in the PLL loop. The advantage of this approach is that inherent filtering is provided by the PLL which can eliminate the need for an RF filter as in the heterodyne case. But this architecture is limited to constant envelope modulation schemes and hence may not be compliant with many recent standards.

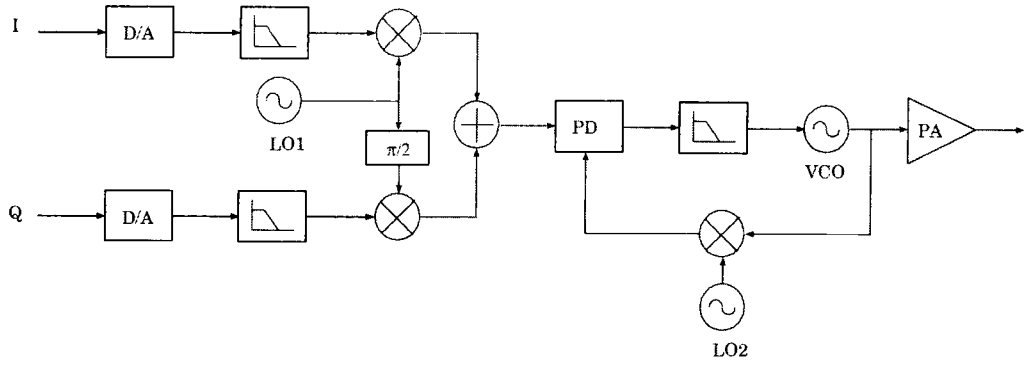


Figure 2.3. PLL-based transmitter.

2.2. Specifications for the Baseband Low-pass Filter

The direct-conversion transmitter architecture was chosen for its relative simplicity in terms of integration onto a single chip and its expected lower power consumption. The baseband reconstruction filter between the D/A converter and the mixer is low-pass and suppresses the image signals and filters out the distortion introduced by the D/A converter. The specifications for the filter are derived from the specifications of the WLAN 802.11b mask and the surrounding blocks. The mask specifications for WLAN 802.11b standard is illustrated in Fig. 2.4.

The critical specification is that the signal level should be -50 dB below the maximum signal strength above a frequency of 22 MHz. Hence, to provide a margin of 10 dB, the filter is designed to have less than -60 dB of gain above 22 MHz. The peak-to-average-ratio (PAR) is 15 dB for OFDM (Orthogonal Frequency Division Multiplexing) and hence, a 1 V_{p-p} signal has an average signal strength of -21 dB. The signal power spectral density (PSD) is -78.4 dBm/Hz considering a filter band-

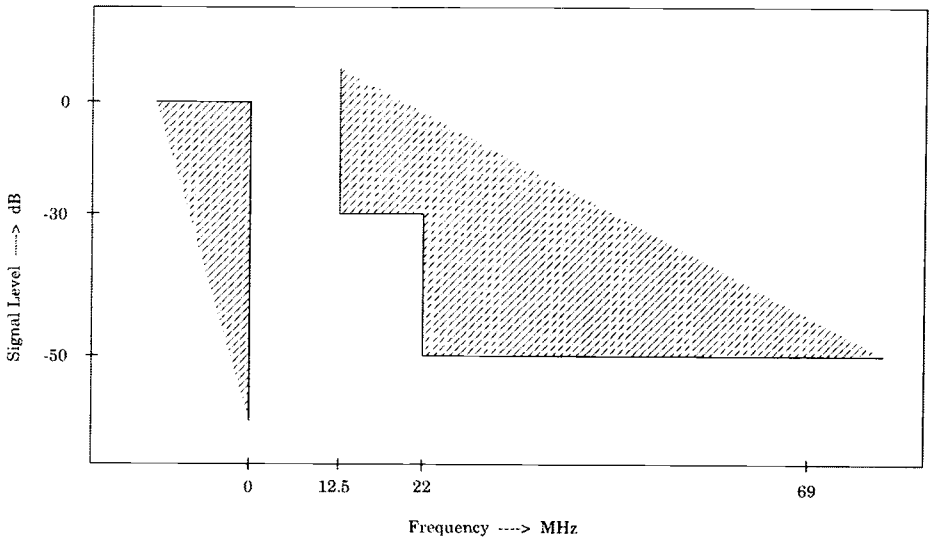


Figure 2.4. Mask specifications for WLAN 802.11b.

width of 11 MHz. The filter is designed for an 8-bit D/A converter with a 80 MHz sampling frequency (f_s). The noise from this D/A converter for a 1 V_{p-p} signal is calculated as:

$$\begin{aligned} \text{D/A Converter Noise} &= \frac{\Delta^2/12}{f_s/2} \\ &= \frac{(1/2^8)^2/12}{80 \times 10^6/2} \\ &= 3.1739 \times 10^{-14}\ \text{V}^2/\text{Hz} \end{aligned} \tag{2.1}$$

where Δ is the least significant bit (LSB) voltage for the D/A converter and is equal to $V_{ref}/2^8$ for an 8-bit converter.

Thus, for a 1 V_{p-p} signal, the noise PSD from the D/A converter is 178.3 $\text{nV}/\sqrt{\text{Hz}}$ or -121.97 dBm/Hz. The signal PSD required at 22 MHz is

-138.4 dBm/Hz which is 60 dB below the maximum signal strength. The attenuation required by the filter at 22 MHz is 16.43 dB. The inherent sample-and-hold operation in the D/A converter suppresses the image signal around the 80 MHz sampling frequency. For a wide-band signal with a bandwidth of 11 MHz, the suppression of the signal at 69 MHz sets the attenuation provided by the filter. The sample-and-hold performs a sinc function in the frequency domain. Thus, the attenuation of the signal at 69 MHz is given by:

$$\begin{aligned} H(s) &= \frac{sin(\pi \times f/f_s)}{\pi \times f/f_s} \\ &= \frac{sin(\pi \times 69 \times 10^6/(80 \times 10^6))}{\pi \times 69 \times 10^6/(80 \times 10^6)} \\ &= 0.1545 \\ &= -16.2 \text{ dB} \end{aligned} \tag{2.2}$$

Therefore, the additional attenuation to be provided by the filter at 69 MHz is $60 - 16.2 = 43.8$ dB.

The filter is designed for an in-band signal-to-noise ratio (SNR) of 40 dB and hence the integrated noise in a 11 MHz bandwidth should be less than -61 dB or 891.3 μV. The minimum out-of-band noise requirement at 22 MHz is equal to the attenuated noise floor of the D/A converter at that frequency which is -138.4 dBm/Hz or 27 nV/$\sqrt{\text{Hz}}$. The filter is also designed to meet a group delay of 50 ns. The filter specifications are summarized in Table 2.1.

Table 2.1. Filter specifications.

Filter Parameter	Specification
Cut-off Frequency	11 MHz
Attenuation > 22 MHz	> 16.43 dB
Attenuation > 69 MHz	> 43.8 dB
Group Delay	< 50 ns
SNR (0-11 MHz)	> 40 dB
Noise Floor > 22 MHz	< 27 nV/$\sqrt{\text{Hz}}$
THD	< -50 dB

3. FILTER DESIGN AND IMPLEMENTATION

3.1. Filter Architecture

3.1.1. Filter Order and Transfer Function

From the filter specifications obtained in Chapter 2, a fourth-order Butterworth filter is chosen with a bandwidth of 12.3 MHz. The frequency response for the low-pass filter simulated in MATLAB is shown in Fig. 3.1.

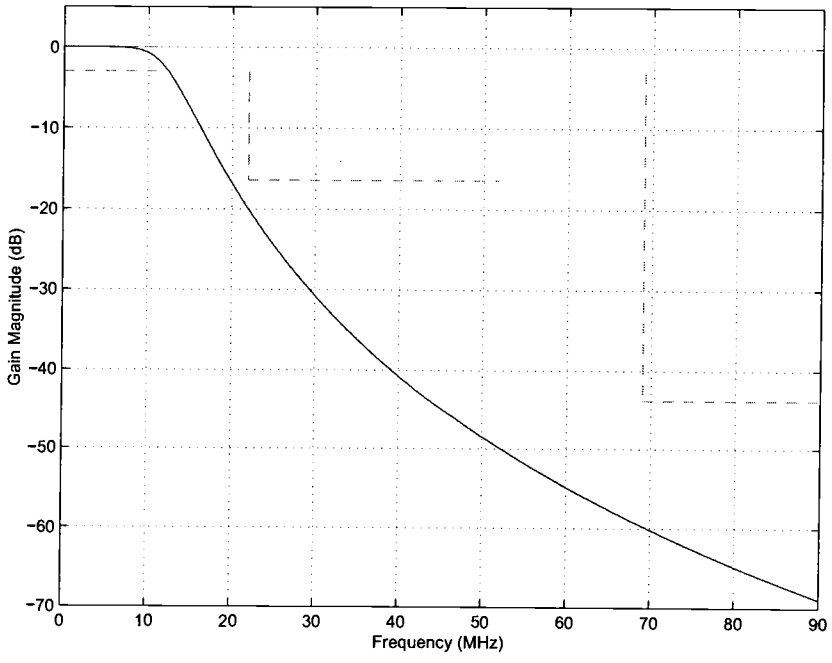


Figure 3.1. Frequency response of the baseband filter with the mask specifications for WLAN 802.11b indicated by the dotted lines.

Simulations show that the Butterworth has a much better group delay then its equivalent Chebyshev and elliptic filters while providing a maximally flat response in the passband. The Butterworth filter shows a group delay of 25 ns while similar orders for Chebyshev and elliptic filters do not meet the group delay specification of 50 ns. The Bessel filter gives a much better group delay but has a higher order and hence it would consume more power and area.

The filter transfer function can be obtained in MATLAB from the specifications and is given by:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{3.659031 \times 10^{31}}{(s^2 + 5.953 \times 10^7 s + 6.049 \times 10^{15})(s^2 + 1.437 \times 10^8 s + 6.049 \times 10^{15})} \tag{3.1}$$

To meet the filter specifications obtained in Chapter 2, the filter order chosen enables the -3 dB bandwidth to fluctuate from 11 MHz to 13.75 MHz (11% variation on either side) compared to a fifth-order filter, which would allow 15% variation from the center frequency but would consume more power and increase the chip area.

3.1.2. Comparison of Filters

There are various architectures of continuous-time filters that can be used to implement the above filter such as the G_m-C filter, the MOSFET-C filter and the active-RC filter. The G_m-C filter has good tunability and can implement higher cut-off frequencies due to the large bandwidth of the transconductors but they have poor linearity, require tuning cicuitry and it is difficult to implement variable gain in the filter [10] [11]. The active-RC filter has much better linearity, it can accommodate

variable gain and it provides increased bandwidths with reduced process technologies [12] [13]. The MOSFET-C filter is tunable with extra circuitry but it has poor linearity compared to the active-RC filter [14] [15]. Hence, given the low supply voltage (1.5 V), the active-RC filter was chosen. The Sallen-Key type of architecture was implemented as it saves considerable power and area compared to the Tow-Thomas or other architectures [13] [16] [17].

3.1.3. Filter Implementation

The fourth-order Butterworth filter implemented consists of two stages, each of which is a single amplifier biquad, shown in Fig. 3.2(a) and Fig. 3.3(a), respectively. The first stage is a high-Q stage followed by a low-Q second stage. The structure of the high-Q stage is similar to that of the low-Q but has an additional positive feedback capacitive loop. This is primarily because the transfer function for the high-Q stage in Fig. 3.2(a), results in lower overall capacitance values than if this loop were not present. The capacitive spread in the first stage due to this high-Q structure is 4 while the total capacitance for this differential stage is 21.4 pF. This sequence of stages, that is the high-Q stage followed by the low-Q stage, minimizes the noise peaking around the -3 dB frequency. Simulations show a 25% improvement in the in-band noise with this sequencing. A differential structure is chosen for better linearity as the even harmonics cancel out, resulting in the elimination of the second harmonic. The primary distortion is due to the third harmonic which is much smaller in magnitude than the suppressed second harmonic.

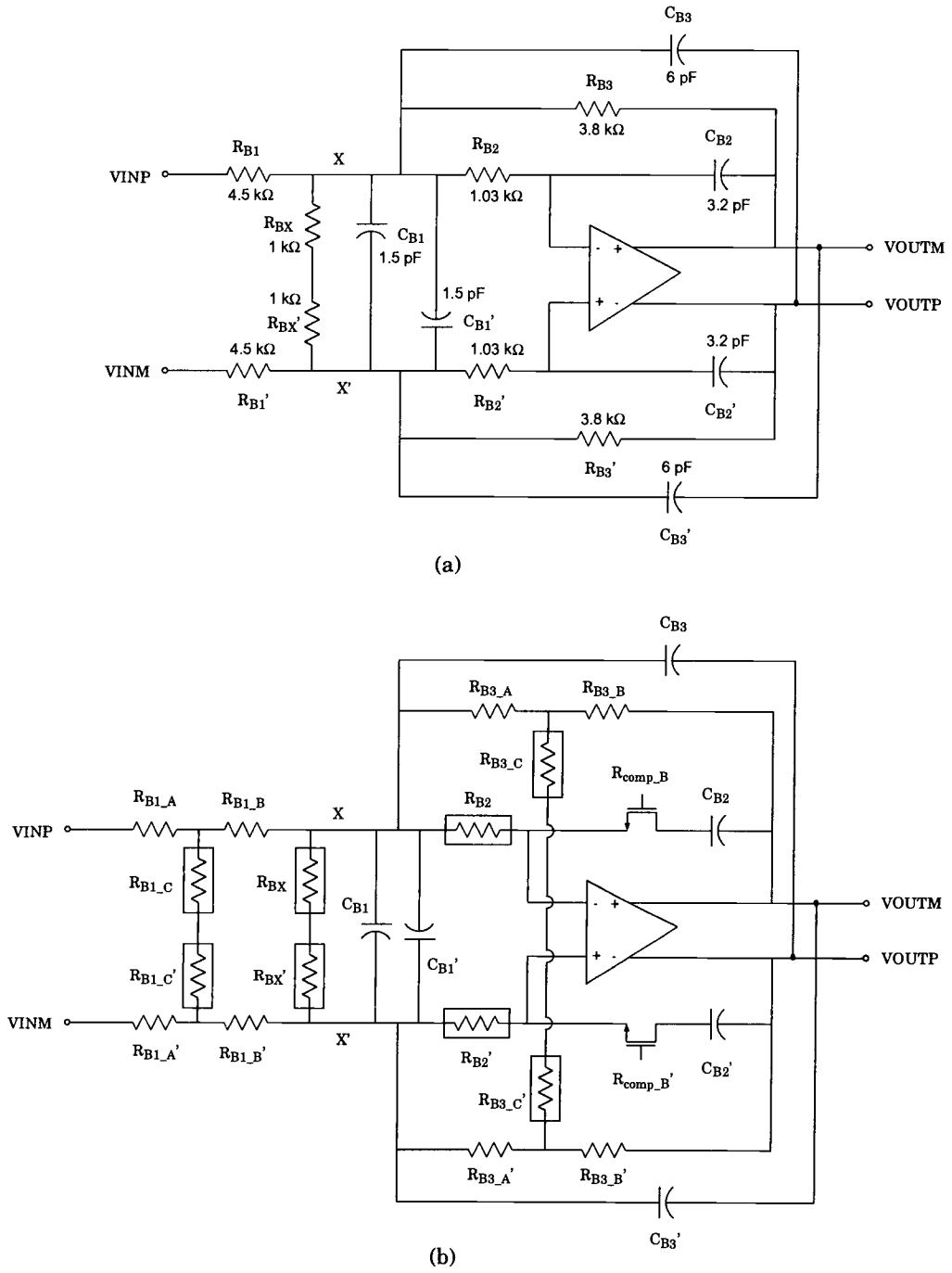
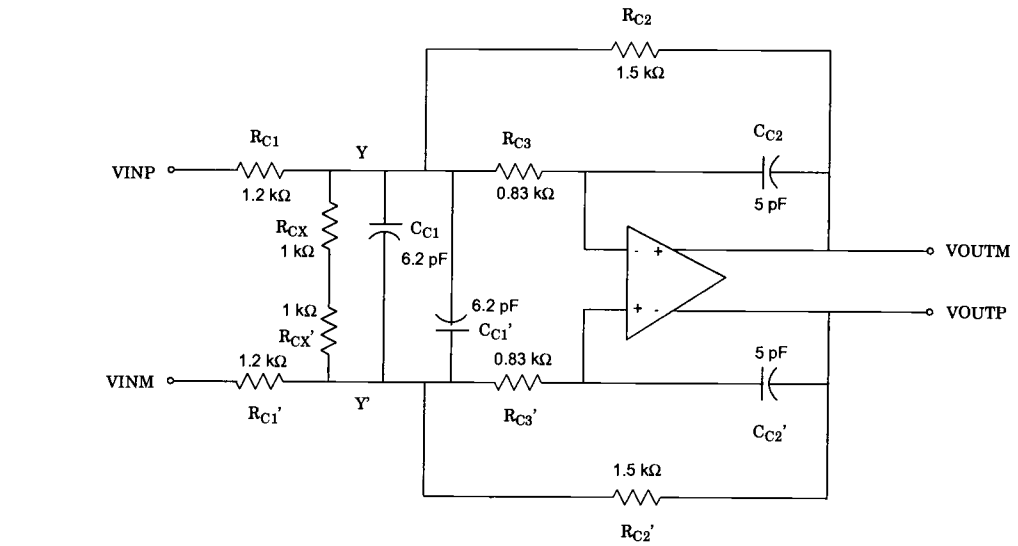
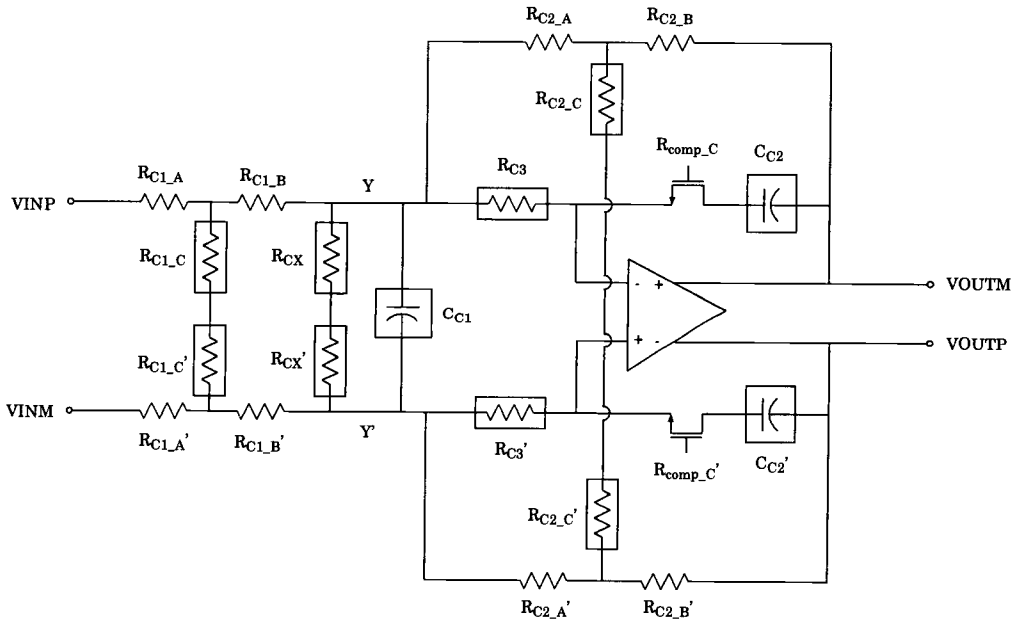


Figure 3.2. First stage biquad. (a) Initial. (b) Modified.



(a)



(b)

Figure 3.3. Second stage biquad. (a) Initial. (b) Modified.

The transfer functions of the circuits in Fig. 3.2(a) and Fig. 3.3(a) and the equations governing the generation of the element values of the filter are given in Appendix A. Element values were adjusted to optimize noise and area while ensuring that the resistive load remains high enough to give a DC opamp gain of at least 80 dB. The signal for WLAN 802.11b is wide-band which implies the presence of multiple tones at different frequencies which have a combined maximum average (or RMS) value, rather than a peak value at one particular tone. Hence, node voltage scaling in the filter is done such that the opamp outputs have their mean square energies set to 0 dB rather than their peak energies.

Trimming of the components needs to be done to account for changes in the component values due to process and temperature variations. Additional trimming is also required to provide variable gain in the filter. This is done by replacing each resistor (or capacitor) by an array of resistors (or capacitors), which have switches connected to one of their terminals. These switches, which are implemented with MOS transistors, are used for selecting a particular combination of resistors (or capacitors). For the single amplifier biquad structures shown in Fig. 3.2(a) and Fig. 3.3(a), some of these switches must be connected to nodes that have relatively high voltage swings. This will cause a variation in the switch resistance with voltage swing, which will introduce distortion. It is usually preferred to connect the switches to common-mode ground nodes since voltage swings at these nodes are minimal.

Frequency trimming for the second stage in Fig. 3.3(a), can easily be implemented with good linearity, using an array of capacitors and switches for C_{C1}, C'_{C1}, C_{C2} and C'_{C2}. This is possible since all the switches in these arrays have one terminal connected to common-mode ground, thus reducing signal swings across them. However, for the first stage shown in Fig. 3.2(a), the positive feedback capacitors,

C_{B3} and C'_{B3} do not have any connections to common-mode ground and neither do resistors, R_{B1}, R'_{B1}, R_{B3} and R'_{B3}. Therefore, some switches, used in this stage for frequency trimming, will experience high voltage swings across their terminals and lead to increased distortion levels as discussed above.

Also, the different DC gains (0 dB to -21 dB) for the filter can be implemented by varying the resistor ratios, R_{C2}/R_{C1} and R'_{C2}/R'_{C1}, in Fig. 3.3(a). This variable attenuation is assigned to the second stage rather than the first stage as the attenuation can then suppress the noise from the first stage. The DC gain of the first stage is fixed at -1.5 dB by the resistor ratios, R_{B3}/R_{B1} and R'_{B3}/R'_{B1}, in Fig. 3.2(a). The variable gain in the second stage, has to be accomplished by replacing the corresponding resistors in Fig. 3.3(a), with resistor arrays and switches. However, R_{C1}, R'_{C1}, R_{C2} and R'_{C2} in Fig. 3.3(a) do not have any connections to common-mode ground. Hence the switches connected to these resistors will also experience high voltage swings across their terminals that lead to increased distortion levels

Thus, the linearity of the filter is degraded due to the use of switches that are connected to nodes with relatively high voltage swings. In order to alleviate this problem, a special hybrid network discussed in Section 3.3 has been used. This hybrid network improves the linearity of the filter while accommodating frequency and variable gain trimming with just two opamp stages. Also, this hybrid network, if implemented with capacitors instead of resistors, results in very high capacitance values, which consumes considerable area. Hence, frequency trimming for the first stage, using this scheme, is implemented using resistors rather than capacitors. The final modified circuit diagrams for the first and second stages are shown

in Fig. 3.2(b) and Fig. 3.3(b), respectively. Resistors (and capacitors) in Fig. 3.2(b) and Fig. 3.3(b), enclosed in boxes, represent trimmed resistor (and capacitor) arrays.

Thus, frequency trimming has been accommodated in the first and second stages using resistive and capacitive arrays, respectively, while gain trimming has been accommodated in the second stage using resistive arrays [18]. The frequency trimming arrays along with the 11% inherent trimming provided by the fourth-order Butterworth filter order accounts for RC variations upto 35%. The filter configured as in Fig. 3.2(b) and Fig. 3.3(b) enables a DC gain variation from 0 dB to -21 dB in steps of -3 dB by using different combinations of resistors for each step. The description of the capacitor and resistor arrays for frequency and gain trimming is described in Section 3.3.

Non-idealities in the amplifiers that are not accounted for, such as finite gain and bandwidth, cause deviations from the ideal transfer function of the filter. Hence, R_{comp_B} and R_{comp_C} , in series with the feedback capacitor in each integrator, respectively, can be used to compensate for the finite gain-bandwidth product of the amplifiers as described in [19]. But a passive implementation of these resistors does not track variations due to process and temperature and simulations show that the resistance values can fluctuate by nearly 50% from the required value. For each integrator, the value of this particular resistor is given by:

$$R_{comp} = \frac{1}{\omega_I C_f} = \frac{C_c}{g_m C_f} \tag{3.2}$$

where ω_I is the unity gain bandwidth of the opamp, C_f is the feedback capacitance of the integrator, g_m is the transconductance of the input transistor of

the opamp and C_c is the internal compensation capacitance from the output of the third stage of the opamp to the output of the first stage of the opamp.

Hence, R_{comp_B} and R_{comp_C} are implemented as NMOS resistors, which can track process and temperature variations if they can inversely track the g_m of the input transistors of the respective amplifiers or if the product, $R_{comp}g_m$, is constant. This is achieved by the tracking circuit shown in Fig. 3.4, which generates a voltage, V_{mosR}, while consuming a minimal amount of current [20]. This voltage, V_{mosR}, drives the gate voltages of R_{comp_B} and R_{comp_C}, respectively.

In Fig. 3.4, if I_{D7} is chosen with respect to $I_{D8}(= 2I_{D9})$ such that $V_{GS5} = V_{GS9}$, and if M4 is biased such that $V_{DS4} = V_{DS8}$, then $V_a = V_b$ and $V_{GScomp} = V_{GS6}$. Since $R_{comp} = 1/(\mu_n C_{ox}(W/L)_{comp}V_{eff\_comp})$ and $g_{m9} = \mu_n C_{ox}(W/L)_9 V_{eff\_9}$, the product, $R_{comp}g_{m9}$, is given by:

$$\begin{aligned} R_{comp}g_{m9} &= \frac{(W/L)_9 V_{eff\_9}}{(W/L)_{comp}V_{eff\_comp}} \\ &= \frac{(W/L)_9 V_{eff\_5}}{(W/L)_{comp}V_{eff\_6}} \\ &= \frac{(W/L)_9}{(W/L)_{comp}}\frac{(W/L)_6}{(W/L)_5} \\ &= constant \end{aligned} \tag{3.3}$$

where, $V_{eff} = V_{GS} - V_{TH}$.

Simulations were done for the filter to analyze effects of temperature variation alone and an average deviation of 0.15 MHz was found at the corner temperatures for the various trimming bit combinations. This is negligible compared to the margin provided by the WLAN mask specifications. For an opamp input-referred offset voltage of 10 mV, the differential output offset voltage was found to be 30 mV.

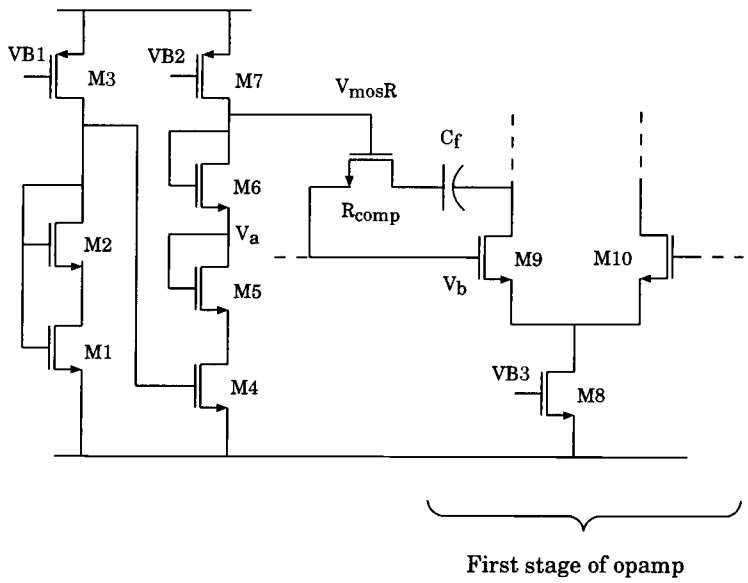


Figure 3.4. Tracking circuitry for the MOS resistor in the integrator feedback path.

3.2. Hybrid Resistor Configuration For Improved Linearity

As mentioned earlier, trimming of certain resistors, in Fig. 3.2(a) and Fig. 3.3(a), require switches that may have relatively large voltage swings at their terminals[1]. To eliminate this problem, a hybrid scheme is used which involves the resistor π-to-Tee network transformation shown in Fig. 3.5, which are related by Eqs. (3.4)-(3.6).

---

[1]In this section, references made to any particular component or node in a figure also implies a reference to the diferential counterpart in that figure.

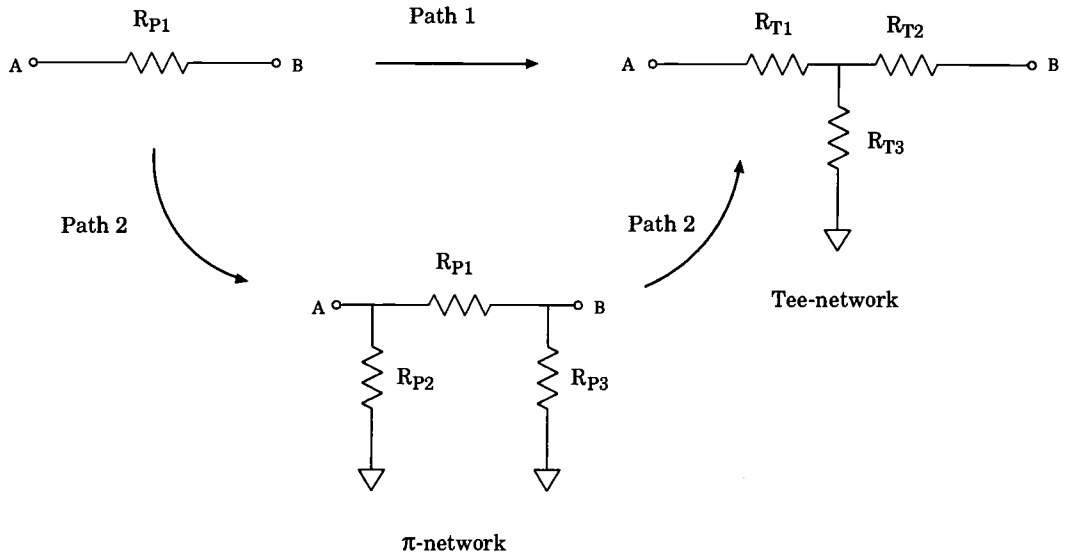


Figure 3.5. Resistor π-to-Tee network transformation.

$$R_{T1} = \frac{R_{P1}R_{P2}}{R_{P1} + R_{P2} + R_{P3}} \tag{3.4}$$

$$R_{T2} = \frac{R_{P1}R_{P3}}{R_{P1} + R_{P2} + R_{P3}} \tag{3.5}$$

$$R_{T3} = \frac{R_{P2}R_{P3}}{R_{P1} + R_{P2} + R_{P3}} \tag{3.6}$$

R_{B1} and R_{B3}, in Fig. 3.2(a), and R_{C1} and R_{C2}, in Fig. 3.3(a), do not have connections to common-mode ground and hence they can be replaced by this Tee-network. This is possible only via the π-network as depicted by Path 2 in Fig. 3.5, since the transformation has to involve 3-port networks for feasibility. For each of these resistors (R_{B1}, R_{B3}, R_{C1} and R_{C2}), R_{P1} in Fig. 3.5 can be set equal to that particular resistor value and the corresponding R_{T1}, R_{T2} and R_{T3} can be calculated using Eqs. (3.4)-(3.6). This transformation enables R_{T3} to be trimmed with the

switches having one terminal to common-mode ground, thus minimizing the distortion introduced by the switches. Hence, R_{B1} and R_{B3}, in Fig. 3.2(a), and R_{C1} and R_{C2}, in Fig. 3.3(a), are replaced by Tee-networks, as shown in Fig. 3.2(b) and Fig. 3.3(b), respectively.

Each Tee-network used requires or produces two additional, unwanted, equivalent π-network resistors, R_{P2} and R_{P3}, as depicted in Fig. 3.5. For the Tee-networks replacing R_{B1}, in Fig. 3.2(a), and R_{C1}, in Fig. 3.3(a), one of the equivalent π-network resistors is connected to the inputs of the respective stages, while for the Tee-networks replacing R_{B3}, in Fig. 3.2(a), and R_{C2}, in Fig. 3.3(a), one of the equivalent π-network resistors is connected to the outputs of the respective stages. Hence, these π-network resistors do not change the transfer functions of the stages since they are connected to the inputs or outputs of the stages. The other π-network resistors are connected to the common nodes, X, in Fig. 3.2(b), and Y, in Fig. 3.3(b), respectively. Therefore, whenever R_{B1_C} and R_{B3_C} is changed to adjust for the process and temperature variations, the value of the resistance connected to node X changes. Similarly, whenever R_{C1_C} and R_{C2_C} is changed to provide different gains, the value of the resistance connected to node Y changes. Hence, additional resistor arrays, R_{BX} and R_{CX}, have been introduced at nodes X and Y, respectively. These additional resistors can now be trimmed to keep the effective resistance at nodes X and Y constant, thus maintaining the same filter transfer function.

3.3. Trimming Arrays

3.3.1. Frequency Trimming

Frequency trimming is done with resistor arrays in the first stage and capacitor arrays in the second stage[2]. For calculation of the element values in the frequency trimming arrays, a variation of ±35% and ±25% in the RC value is considered. For each bit sequence, the combination of capacitors and resistors selected in the respective arrays, bring the bandwidth value back to within ±10%, which is within the mask specifications as discussed earlier.

Fig. 3.6 shows a typical resistor array used for frequency or gain trimming. Eqs. (3.4)-(3.6) are directly used while calculating the nominal process corner values for resistor arrays, R_{B1_C} and R_{B3_C}, in Fig. 3.2(b). For other frequency trimming bit combinations, the initial values in these resistor arrays are calculated assuming that the resistors, R_{T1}, R_{T2} and R_{T3}, in every Tee-network, in Fig. 3.2(b) and Fig. 3.3(b) have changed by the particular RC product variations (±25% or ±35%). Thus each bit combination adjusts the value to within 10% of its nominal value. This applies to the equivalent π-network resistor, R_{P1}, for every Tee-network, in Fig. 3.2(b) and Fig. 3.3(b) and the resistances at nodes X and Y in these figures. Thus, it corrects these RC product variations. This is within the 11% deviation allowed from the nominal designed value of 12.3 MHz bandwidth as mentioned in Section 3.1.

---

[2]In this section, references made to any particular component or node in a figure also implies a reference to the diferential counterpart in that figure.

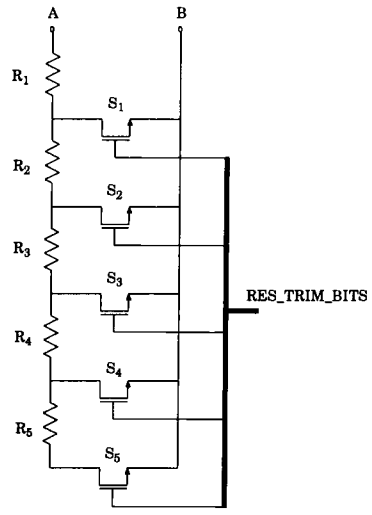


Figure 3.6. Typical resistor array.

The final values of the resistors in the array are chosen to optimize area in the following manner. In Fig. 3.6, R_1 is chosen to be the smallest resistor value for any bit combination which is activated by having only switch S_1 on. The next smallest resistor value for any of the remaining bit combinations is set to $(R_1 + R_2)$ which is activated by having only switch S_2 on. This algorithm is continued for the remaining bit combinations.

A typical capacitor array used for frequency trimming is shown in Fig. 3.7. The bit combination for the nominal process corner is set to $(C_1 + C_2 + C_3) = C_{nom}$. The other bit combinations are represented by Eqs. (3.7)-(3.10) which indicate the combination of parallel capacitors used (to optimize area) for each process corner variation considered ($\pm 25\%$ or $\pm 35\%$). Capacitor values for each array can be obtained by solving these equations.

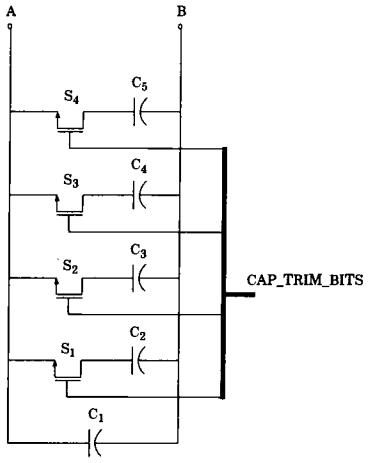


Figure 3.7. Typical capacitor array.

$$(1+\frac{35}{100})C_1 = (1+\frac{10}{100})C_{nom} \tag{3.7}$$

$$(1+\frac{25}{100})(C_1+C_2) = (1+\frac{10}{100})C_{nom} \tag{3.8}$$

$$(1-\frac{25}{100})(C_1+C_2+C_3+C_4) = (1-\frac{10}{100})C_{nom} \tag{3.9}$$

$$(1-\frac{35}{100})(C_1+C_2+C_3+C_4+C_5) = (1-\frac{10}{100})C_{nom} \tag{3.10}$$

It can be observed from Eqs. (3.7)-(3.10) that the frequency trimming corrects the capacitance value to within 10% of the nominal capacitance value.

3.3.2. Gain Trimming

As mentioned earlier, each bit combination sets the resistor values to give a gain between 0 dB and -21 dB in steps of -3 dB. It is to be noted that R_{C3}, in

Fig. 3.3(b), is also trimmed for different gains in order to keep the ω_o and Q of the circuit intact for all gains. For resistors, R_{C1} and R_{C2}, in Fig. 3.3(a), Eqs. (3.4)-(3.6) can be used to find their respective transformed Tee-network resistors, in Fig. 3.3(b). R_{C1_A}, R_{C1_B}, R_{C3_A} and R_{C3_B}, in Fig. 3.3(b), are kept fixed, while R_{C1_C}, R_{C3_C} and R_{CX} are trimmed. Each gain trimming bit combination sets R_{C1}, R_{C2}, R_{C3} and the resistances at Y, in Fig. 3.3(a) to their respective values for a particular gain. The algorithm for the actual resistor values in the resistor arrays of Fig. 3.3(b) is similar to that for frequency trimming.

3.4. Operational Amplifier

The operational amplifier is the most important block in this filter as it is the main contributor to the linearity of the filter [13]. The finite gain and bandwidth of the opamp also produces deviations from the ideal characteristics of the filter transfer function. The opamp designed has three stages with nested-Miller compensation and is shown in Fig. 3.8.

The three stages in the amplifier together give high linearity performance due to the high gain and the reduction of distortion internal to the opamp by the two nested feedback loops. The first and second stages are differential amplifiers while the last stage is a differential class A common source stage. The input devices in all the stages are NMOS in order to maximize the transconductance and reduce the thermal noise which is the main contributor to the out-of-band noise at 22 MHz. The overall DC gain of the amplifier with a 1 KΩ and 5 pF load is 80 dB while the unity gain bandwidth and phase margin are 500 MHz and 60°, respectively. The

DC current consumption is 3.8 mA with a single 1.5 V supply voltage where each differential branch of the last stage consumes 1.05 mA.

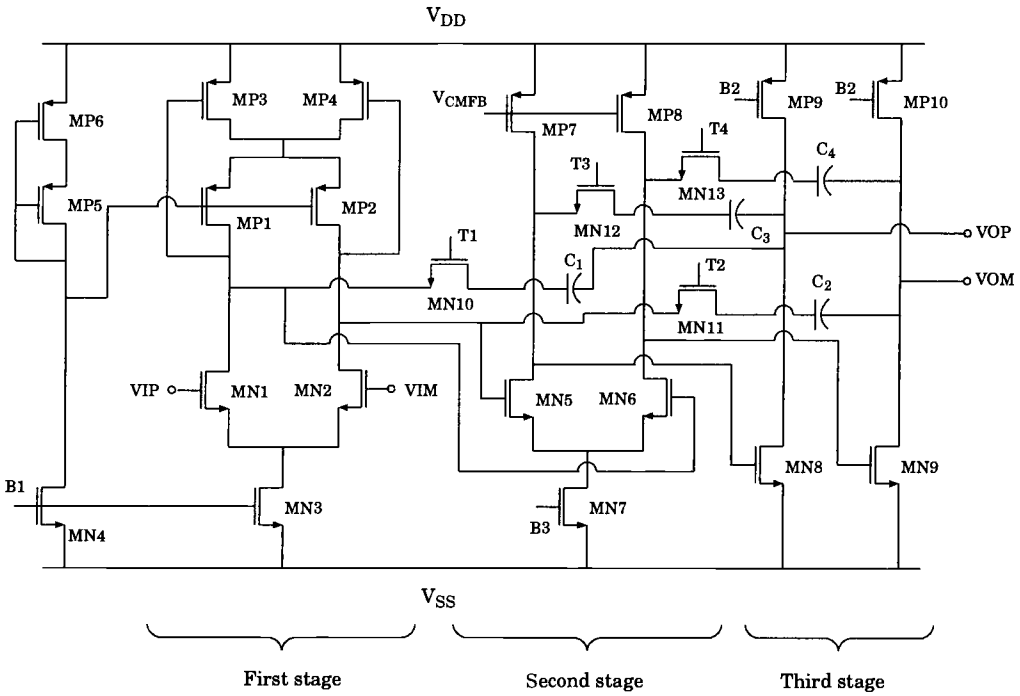


Figure 3.8. Three-stage operational amplifier.

The first stage has PMOS transistors, MP3 and MP4, biased in the triode region. These PMOS transistors form part of a local common-mode feedback for the first stage. They act as MOS resistors whose gates are driven by the first stage output voltage. Thus any variation in the output common-mode voltage changes the resistance of these PMOS transistors and hence the voltage drop across them.

This forces the output common-mode voltage to pull back to its original value of 0.75 V and stabilizes it.

The common-mode feedback circuitry for the second and third stages is shown in Fig 3.9. The output common-mode voltage is tapped with the passive resistive network, R_{a1} and R_{a2}. Each resistor has a large value, in this case 10 KΩ, to minimize additional loading of the opamp output node. This common-mode voltage is then compared with a reference voltage generated by the two PMOS transistors, MP11 and MP12, which form a voltage divider network. The difference is amplified by the single-stage differential amplifier which has diode-connected loads for low gain and hence a lower gain-bandwidth product. This helps in stabilizing the common-mode feedback network. The output from this differential amplifier is then fed to the second stage of the main amplifier, via the V_{CMFB} voltage on the gates of MP7 and MP8 in Fig. 3.8. Thus the common-mode feedback for the second and third stages is provided by this one circuit.

The main opamp is stabilized with nested-Miller compensation loops and each feedback branch consists of a MOS resistor and capacitance in series. MN10 and C_1, MN11 and C_2, MN12 and C_3, MN13 and C_4, in Fig. 3.8, are the four feedback branch pairs forming the nesting-Miller compensation loops. These MOS resistors are biased by circuitry similar to that of Fig. 3.4 for R_{comp} to provide pole-zero tracking with process and temperature variations [23]. Fig. 3.13 shows the actual biasing circuit for these MOS resistors, MN10, MN11, MN12 and MN13 in Fig 3.8 via nodes T1, T2, T3 and T4, respectively. This compensation scheme for the opamp was found to be the most optimum of the three nested-Miller compensation schemes published recently [22] [23], in terms of bandwidth, distortion and tracking with process or temperature variation. These three nested-Miller compensation schemes

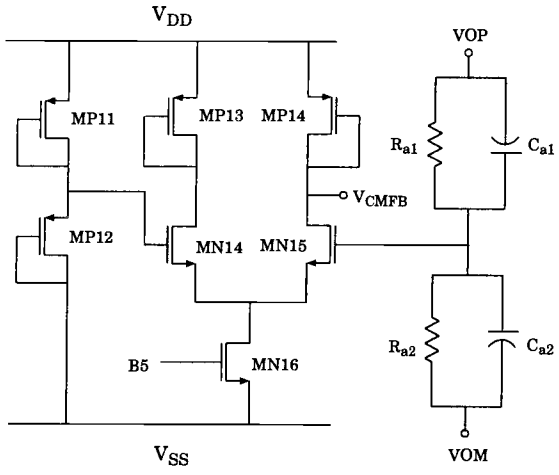


Figure 3.9. Common-mode feedback for the opamp second and third stages.

are shown in Fig. 3.10, Fig. 3.11 and Fig. 3.12, where VO1, VO2 and VO3 are the outputs of the first, second and third stages, respectively.

Table 3.1 compares the three compensation schemes in terms of their bandwidths, phase margins and component values. Table 3.2 compares the tracking performance of these compensation schemes with variations in process and temperature when MOS resistors are used. In Table 3.2, the weak corner refers to a weak process at 125^o C, the nominal corner refers to a nominal process at 27^o C and the strong corner refers to a strong process at -40^o C.

From Fig. 3.10, Fig. 3.11 and Fig. 3.12, it can be seen that compensation scheme 1 is the simplest of the three schemes to implement. This is because it has has the least number of components and hence the interdependence of the component values on each other is reduced for pole-zero cancellation [22]. But Table 3.1 shows

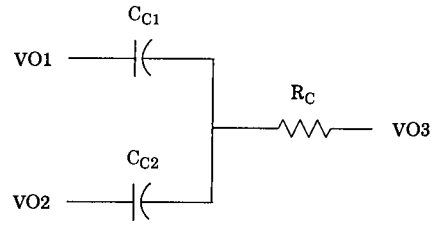


Figure 3.10. Compensation scheme 1.

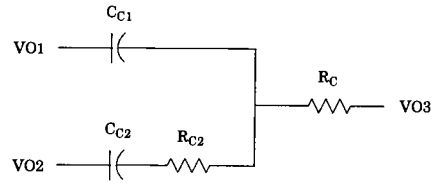


Figure 3.11. Compensation scheme 2.

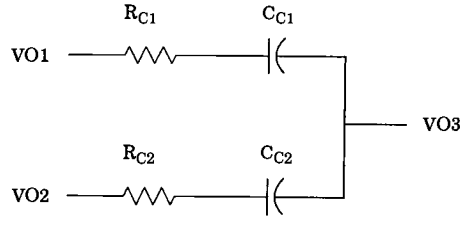


Figure 3.12. Compensation scheme 3.

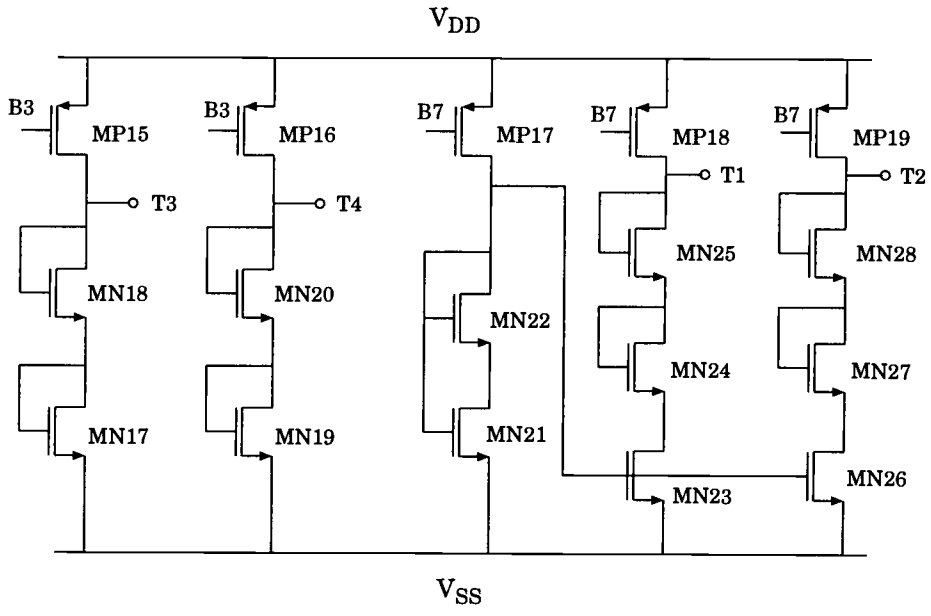


Figure 3.13. Biasing circuit for the compensation MOS resistors in the opamp.

Table 3.1. Comparison between three R-C internal feedback compensation schemes.

Scheme	UGBW	PM	Gain(10 MHz)	C_{C1}	C_{C2}	$R_{C1}(R_C)$	R_{C2}
Scheme1	500 MHz	70^o	32 dB	2.3 pF	3.3 pF	200 Ω	
	550 MHz	60^o	34 dB	2 pF	2.8 pF	200 Ω	
Scheme2	645 MHz	70^o	36 dB	1.6 pF	1 pF	110 Ω	530 Ω
	930 MHz	61^o	40 dB	1.1 pF	1.3 pF	130 Ω	600 Ω
Scheme3	560 MHz	73^o	34 dB	2 pF	1.5 pF	300 Ω	500 Ω
	758 MHz	61^o	38 dB	1.2 pF	1.5 pF	400 Ω	500 Ω

that it requires higher capacitance values and has a smaller achievable unity gain bandwidth than the other two schemes.

Table 3.2. Comparison of process variation performance for the compensation schemes.

Sch.	Corner	With Ideal Resistor				With MOS Resistor			
		BW	PM	%Δ BW	%Δ PM	BW	PM	%Δ BW	%Δ PM
1	Weak	250 MHz	44^o	-51%	-37%	328 MHz	57^o	-38%	0%
	Nom.	506 MHz	70^o	-	-	530 MHz	57^o	-	-
	Strong	1.1 GHz	44^o	+100%	-37%	920 MHz	40^o	+73%	+30%
2	Weak	420 MHz	49^o	-49%	-30%	506 MHz	55^o	-28%	-9%
	Nom.	645 MHz	70^o	-	-	712 MHz	61^o	-	-
	Strong	1 GHz	79^o	+59%	-13%	1.1 GHz	61^o	+60%	0%
3	Weak	345 MHz	50^o	-37%	-32%	450 MHz	67^o	-22%	0%
	Nominal	556 MHz	73^o	-	-	583 MHz	67^o	-	-
	Strong	961 MHz	79^o	+72%	+8%	804 MHz	66^o	+37%	-1%

Table 3.1 also shows that compensation scheme 2 has the lowest capacitance values and the highest achievable unity gain bandwidth. As mentioned earlier in this section, MOS transistors can be used for the resistors in these compensation schemes to provide tracking with process and temperature variations. In compensation scheme 2 however, the resistor connected at the opamp output has high voltage swings across its terminals. Hence, it cannnot be implemented using a MOS transistor because of the high distortion introduced. Alternatively, tracking using poly

resistors, with compensation scheme 2, can be achieved by using the biasing circuitry shown in Fig. 3.14.

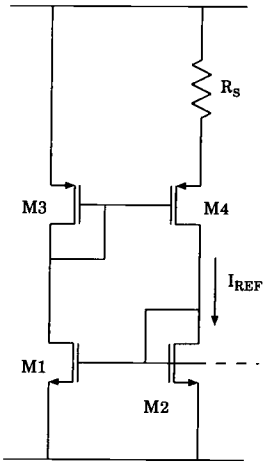


Figure 3.14. Biasing circuit for tracking an on-chip resistor.

In Fig. 3.14, assuming M1 and M2 have identical (W/L) ratios, I_{REF} is given by:

$$\begin{aligned} I_{REF} &= \frac{2}{\mu_p C_{ox}}[\frac{1}{\sqrt{(W/L)_3}} - \frac{1}{\sqrt{(W/L)_4}}]^2 \frac{1}{R_s^2} \\ &= constant \times \frac{1}{R_s^2} \\ &\alpha \ \frac{1}{R_s^2} \end{aligned} \tag{3.11}$$

If the currents in all branches of the main amplifier are mirrored from I_{REF} and assuming that the ratio of all resistances in the circuit is a constant, we have:

$$g_{m\_in} \ \alpha \ \sqrt{I_{in}} \ \alpha \ \sqrt{I_{REF}} \ \alpha \ \frac{1}{R_s} \ \alpha \ \frac{1}{R_z}$$

where, g_{m_in} is the transconductance of the input transistor of any particular stage of the opamp, I_{in} is the current flowing through the input transistor of that stage and R_z is the nested-Miller compensation resistance in the feedback branch associated with that particular input transistor. Thus, the transconductance of the input transistors of all the stages in the opamp inversely track a single on-chip resistor and hence also track the respective nested-Miller compensation resistances. Thus, pole-zero tracking with process and temperature variation is achieved.

However, on-chip poly resistors in integrated circuit processes may fluctuate by $\pm 25\%$ due to process and temperature variations and simulations show that the latter scheme leads to nearly doubling of the power consumption with such resistor variations. Compensation scheme 3 was found the most suitable since it provides a large enough bandwidth and good pole-zero tracking with the use of a MOS resistor and it is not limited by the constraints of compensation scheme 2. The distortion performance of compensation schemes 2 and 3 are tabulated in Appendix B.

A band-gap reference current source is used as the primary current source for the opamp and this current is then mirrored to all the different branches in the opamp.

The small-signal model for a single-ended version of the three-stage opamp of Fig. 3.8 is shown in Fig. 3.15. Neglecting the parasitic capacitances and certain second-order terms, the frequency dependent gain of the opamp is given by:

$$A(s) = A_o \frac{1 + [R_{C1}C_{C1} + (R_{C2} - \frac{1}{g_{m3}})C_{C2}]s - \frac{1+(1-g_{m3}R_{C2})g_{m2}R_{C1}}{g_{m2}g_{m3}} C_{C1}C_{C2}s^2}{(1 + \frac{s}{w_{p1}})[1 + (R_{C2} + \frac{1}{g_{m2}} - \frac{1}{g_{m3}})C_{C2}s + \frac{C_{C2}C_L}{g_{m2}g_{m3}}s^2]} \tag{3.12}$$

where, g_{mi} and r_{oi} are the transconductances and output resistances of the i-th stage respectively, A_o is the open-loop DC gain and is equal to

$g_{m1}g_{m2}g_{m3}r_{o1}r_{o2}r_{o3}$, while w_{p1} is the dominant pole frequency and is approximately equal to $1/(r_{o1}g_{m2}r_{o2}g_{m3}r_{o3}C_{C1})$.

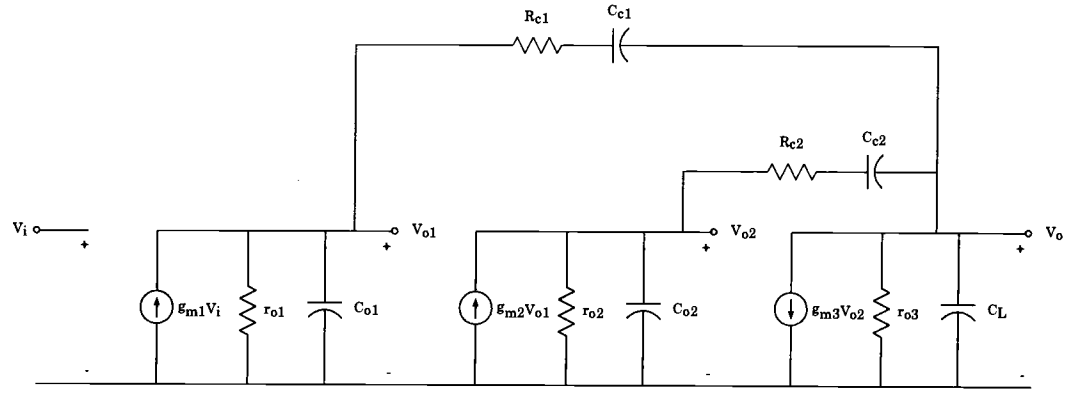


Figure 3.15. Small-signal model of a three-stage opamp.

Eq.(3.12) also contains two other higher order poles and two zeroes. Initially, neglecting the compensation resistors, we can choose appropriate values for the compensation capacitors to push these two poles beyond the unity-gain frequency. Then R_{C2} and R_{C1} can be chosen to eliminate the s^2 term in the numerator and make the s term positive to create a LHP zero. This improves the phase margin of the amplifier. The relationship between R_{C1} and R_{C2} is given by :

$$R_{C2} = \frac{1}{g_{m2}g_{m3}R_{C1}} + \frac{1}{g_{m3}} \tag{3.13}$$

3.5. Other Blocks and Issues

Three bit frequency and gain trimming has been used and a digital logic block, consisting of gates with a dual 3V supply voltage, has been designed to provide the control signals for the various resistive and capacitive arrays. The higher power supply provides better linearity for the switches. The diagrams depicting this logic is shown in Appendix C and the layout of this block is shown in Appendix E.

The circuit diagrams for the opamp and MOS resistors from CADENCE (with the (W/L) values for all transistors) are shown in Appendix D. The MATLAB code written for the generation of the initial element values from the transfer function and the final element values after the π-to-Tee transformation, including the resistor and capacitor values in each of the individual arrays, is given in Appendix F.

4. LAYOUT AND MEASUREMENTS

4.1. Layout

The filter was fabricated in a 0.15 μm 1.5 V 3 V CMOS technology featuring a single poly layer and six metal layers. The layout of the differentially implemented filter (excluding the digital logic block) is shown in Fig. 4.1. The two stages were cascaded one after the other for easier connections between the two stages and from each stage to the other blocks in the chip. For the differential implementation of this filter, care was taken that all components were symmetrically placed to reduce mismatches between the differential counterparts. The total area consumed by this filter is 726 $\mu m \times 290$ μm.

Poly (with silicide base) resistors are used as they have a very high sheet resistance (180 $\Omega/\ \square$) and hence consume minimal area. Parallel sections connected together have been used in preference to serpentine structures to minimize mismatches. Also, resistors with very small resistance values have been implemented using parallel resistors with higher resistance values to improve relative accuracy and make the contact resistance negligible compared to the resistors. The layout contains metal-metal flux capacitors having a capacitance density of 1.4 fF/μm^2. These capacitors make use of the fringe capacitance between the two metal layers that act as the plates of the capacitor. These capacitors are laid over a n-well which is grounded for better isolation from the substrate. Dummy resistors and fingers have been used for the resistors and switches, respectively, to minimize mismatches due to the etching process and care has been taken to ground these dummies to prevent any pick up of noise by floating nodes.

For differential layout matching considerations, the resistors are placed closer to their differential counterparts compared to the series switches, which have a much lower resistance value. Care has been taken to avoid metal routing over components in order to prevent signal coupling to these metal layers. Wherever routing has been unavoidable over components, an additional metal layer, which is grounded, has been used for separation between the routing metal and the component to absorb any coupling signals, thus acting as a shield. A considerable number of substrate contacts are placed around most of the blocks to minimize the amount of signal coupling through the substrate.

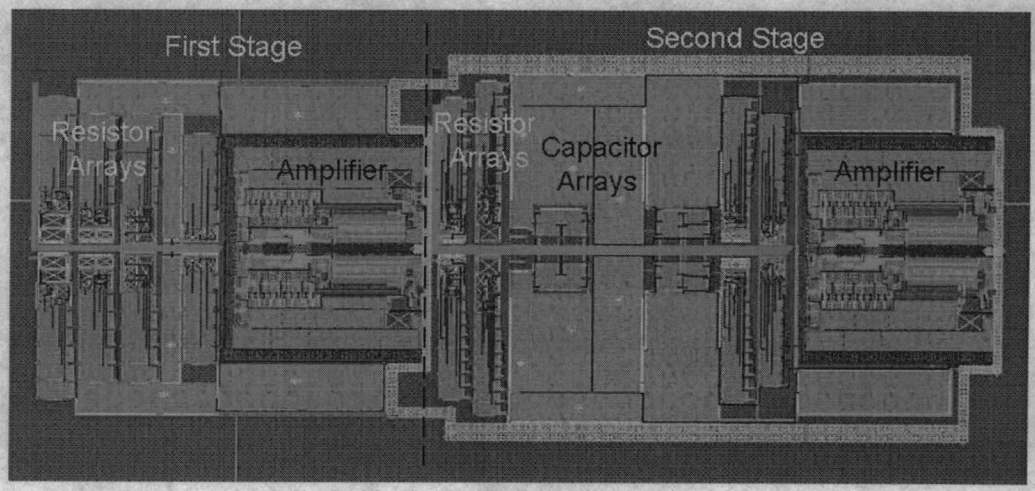


Figure 4.1. Layout of the filter.

4.2. Testing and Measurements

The system was tested with a 1.5 V and a 3.3 V power supply and the die photo of the filter is shown in Fig. 4.2. The filter power dissipation was measured to be 12.2 mW.

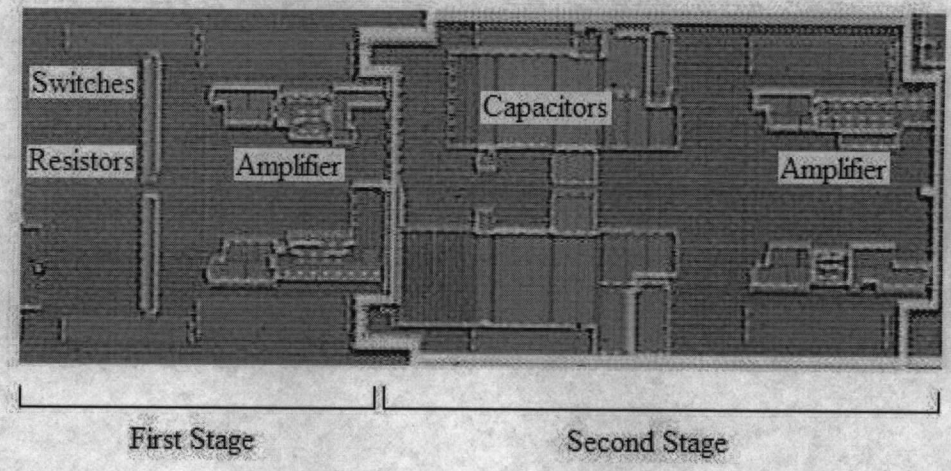


Figure 4.2. Die photo of the filter.

The measured frequency response of the filter is shown in Fig. 4.3 with a -3 dB cut-off frequency of 9 MHz. The total harmonic distortion (THD) is measured for differential input sine waves applied at 1 MHz and 5 MHz. This measurement is independently conducted for 1.2 V_{p-p}, 1.6 V_{p-p} and 2 V_{p-p} signals. Fig. 4.4 shows the frequency spectrum of the output signal for a 2 V_{p-p} input signal at 1 MHz, from which a THD of -72 dB is obtained.

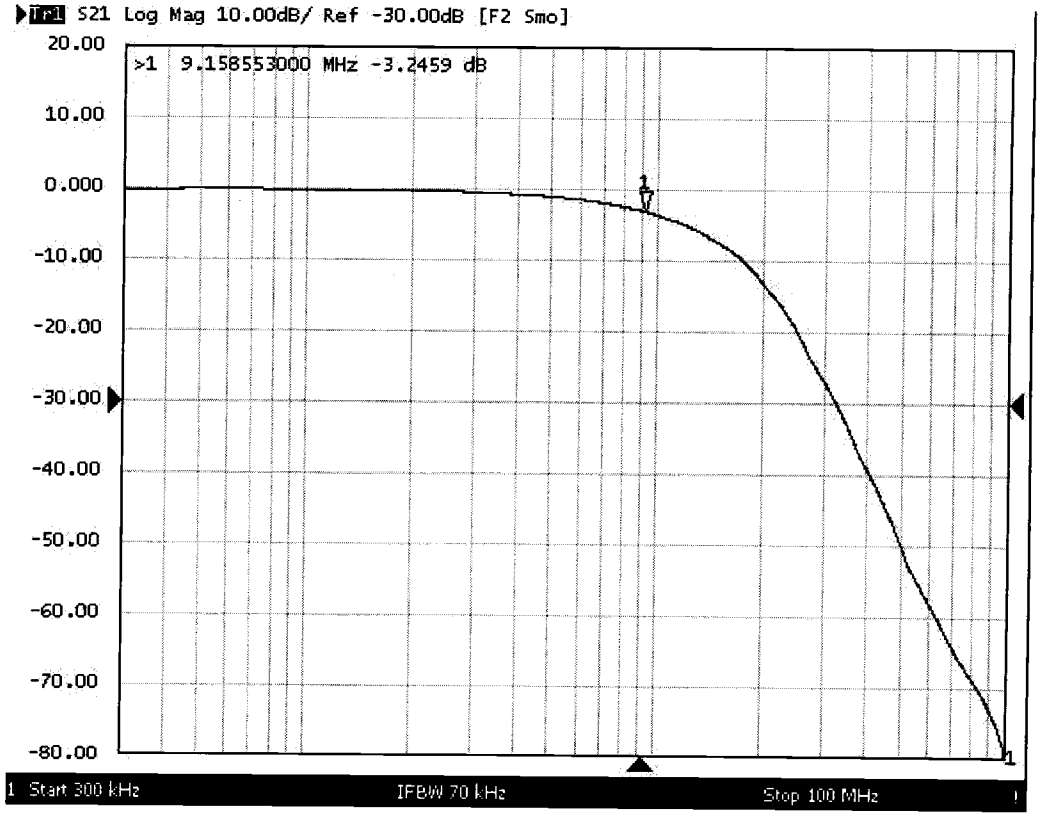


Figure 4.3. Frequency response of the filter.

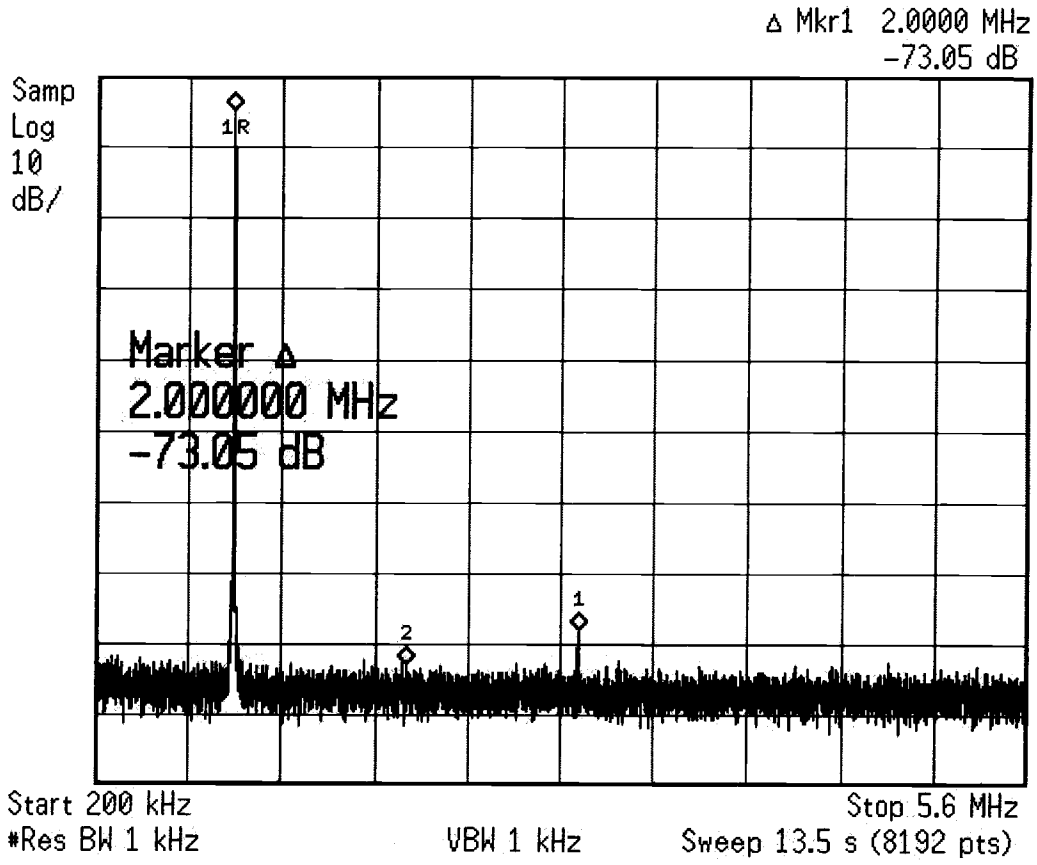


Figure 4.4. Frequency spectrum of the output signal for a 2 V_{p-p} input sine wave applied at 1 MHz.

The third-order intermodulation distortion is also measured by applying two tones centered around 1 MHz. These tones are located at frequencies of 0.9 MHz and 1.1 MHz and measurements are independently conducted for 1.6 V_{p-p} and 2 V_{p-p} signals. It is found that the third-order intermodulation product is -69 dB below the input signal tones for a 2 V_{p-p} signal as observed in Fig. 4.5.

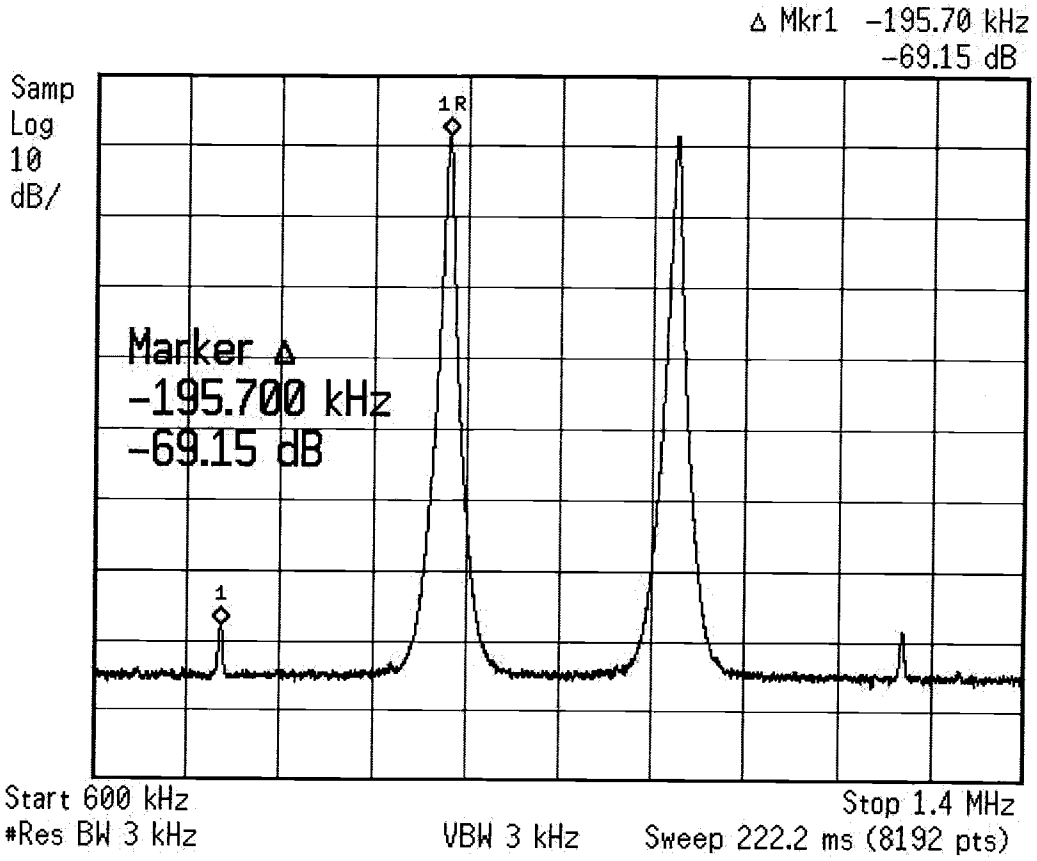


Figure 4.5. Frequency spectrum of the output signal for a 2 V_{p-p} input signal, with tones applied at 0.9 MHz and 1.1 MHz.

The measured in-band SNR assuming a 2 V_{p-p} signal is 62 dB and the measured noise floor at 22 MHz is 24 nV/$\sqrt{\text{Hz}}$. The gain trimming varies the gain in steps of around -3 dB with a maximum step error of 0.6 dB. The -3 dB frequency could be decreased by the bandwidth trimming scheme but could not be increased as one of the bits in the digital logic was not connected properly in the layout. The simulated and measured results of the filter are summarized in Table 4.1. The filter is connected via a MOS switch to an on-chip buffer. The output of this buffer is then tapped for conducting measurements. The differences in the measured and simulated results are attributed to the switches at the output of the filter, the distortion limits of the buffer and the layout asymmetries and parasitics.

A comparison of this work with recently published work on continuous-time filters is tabulated in Table 4.2. As each of these filters have a different order and -3 dB frequency, a figure of merit for the power dissipation is used for comparison. This normalized power dissipation (in mW/pole/MHz) is calculated as:

$$Normalized\ Power\ Dissipation = \frac{Actual\ Power\ Dissipation\ (mW)}{No.\ of\ Poles \times -3dB\ Frequency\ (MHz)} \tag{4.1}$$

The total area consumed by each filter is given. The noise performance is compared after normalization to a power spectral density (in nV/$\sqrt{\text{Hz}}$), rather than the signal-to-noise ratios. This is because each filter has different maximum signal swings and different signal bandwidths. For a good comparison of linearity, three signal tones at frequencies of 0.1 MHz, 1 MHz and 5 MHz have been selected. The measured total harmonic distortion at these three frequency tones and the third order intermodulation products at 1 MHz are given. The ratio of the peak-to-peak

output voltage swing to supply voltage has also been specified for each of these measurements, as an accurate measure of comparison.

Designing with low power supply voltages typically results in reduced overall performance. However, with the improvements described here, the performance exceeds that of previously published work. The THD and IM3 exceed that of each of the previous designs. This is achieved with a peak-to-peak voltage to supply ratio of nearly 1 and a signal frequency of 1 MHz while the next closest design has a ratio of 0.6 and a signal frequency of 0.1 MHz.

Table 4.1. Summary of simulated and measured results.

Parameters	Simulated Results	Measured Results
Technology	0.15 μm CMOS	0.15 μm CMOS
Filter Power Consumption	11.4 mW	12.2 mW
Supply Voltage	1.5 V	1.5 V
Active Area	726 μm $\times$ 290 μm	726 μm $\times$ 290 μm
Cut-off Frequency	11 MHz	9 MHz
SNR (2 V_{p-p} and 0-11 MHz BW)	67 dB	62 dB
THD (1.6 V_{p-p} at 1 MHz)	-77 dB	-74 dB
THD (2 V_{p-p} at 1 MHz)	-75 dB	-72 dB
THD (1.6 V_{p-p} at 5 MHz)	-71 dB	-61 dB
THD (2 V_{p-p} at 5 MHz)	-67 dB	-55 dB
IM3 (1.6 V_{p-p} around 1 MHz)	-78 dB	-71 dB
IM3 (2 V_{p-p} around 1 MHz)	-75 dB	-69 dB

Table 4.2. Comparison with recently published work.

Ref.	Tech.	Norm. P_d (mW/pole /MHz)	Power Supply (V)	Active Area (mm^2)	-3dB Freq. (MHz)	Noise (nv/ $\sqrt{\text{Hz}}$)	Linearity			
							f_{in} (MHz)	$\frac{V_{p-p}}{V_{supp}}$	THD (dB)	IM3 (dB)
[25]	BiCMOS	3.16	2.5	1.15	0.6	140	0.1	0.8	-49	-
[26]	CMOS	17.55	3.3	1.53	0.49	500	0.1	0.6	-75	-
[27]	CMOS	0.8	1	3.61	1	120	0.1	0.65	-56	-
							1	0.32	-	-48
[28]	BiCMOS	19.63	10	25.83	8	742	1	0.5	-	-65
[29]	CMOS	0.3	2.5	3	100	-	5	0.18	-46	-
[12]	CMOS	2.78	3.3	-	16.5	33.6	5	0.85	-60	-
This Work	CMOS	0.33	1.5	0.21	9	68	1	0.8	-76	-
							1	1.07	-74	-71
							1	1.33	-72	-69
							5	0.8	-64	-
							5	1.07	-61	-
							5	1.33	-55	-

5. CONCLUSIONS

In this thesis, the design of a variable gain, high linearity, low power continuous-time filter for WLAN applications in a 1.5 V 3 V 0.15 μm CMOS process has been described. This fourth-order low-pass filter is introduced between the D/A converter and the mixer in the transmit channel and has a measured bandwidth of 9 MHz while achieving high linearity for signals within this bandwidth.

This filter uses single amplifier biquads for lower power consumption but reduces the distortion in these structures, introduced by voltage swings across switches used for frequency and gain trimming. This is achieved by using a special π-to-Tee transformation network for the resistor arrays. Three stage opamps with nested-Miller compensation are used to improve the frequency response and linearity of the filter. The frequency response is also improved with a tracking mechanism for the resistor in series with the integrator feedback capacitors.

BIBLIOGRAPHY

[1] A.P. Chandrakasan, S.Sheng, and R.W. Brodersen, "Low power CMOS digital design," *IEEE J. Solid-State Circuits,* vol. 27, pp. 473-484, Apr. 1992.

[2] C.S. Wong, "A 3-V GSM baseband transmitter," *IEEE J. Solid-State Circuits,* vol. 34, pp. 725-730, May 1999.

[3] A. Rofougaran, G. Chang, J.J. Rael, J.Y.-C. Chang, M. Rofougaran, P.J. Chang, M. Djafari, M.-K. Ku, E.W. Roth, A.A. Abidi, and H. Samueli, "A single-chip 900-MHz spread-spectrum wireles transceiver in 1-μm CMOS - Part *I*: Architecture and transmitter design," *IEEE J. Solid-State Circuits,* vol. 33, pp. 515-534, Apr. 1998.

[4] T.H. Lee and H. Samavati, and H.R. Rategh,"5-Ghz CMOS wireless LANs," *IEEE Trans. Micr. Theory Tech.,* vol. 50, pp. 268-280, Apr. 2002.

[5] A.A. Abidi, "Direct-conversion radio transceivers for digital communication," *IEEE J. Solid-State Circuits,* vol.30, pp. 1399-1410, Dec. 1995.

[6] M. Borremans, M., M. Steyaert, and T. Yoshitomi, "A 1.5 V, wide band 3 GHz, CMOS quadrature direct up-converter for multi-mode wireless communications," *Proc. IEEE Cust. Int. Circuits Conf.,* pp. 79-82, May 1998.

[7] C. Marshall, F. Behbahani, W. Birth, A. Fotowai, T. Fuchs, R. Gaethke, E. Heimeri, S. Lee, P. Moore, S. Navid, and E. Saur, "A 2.7 V GSM transceiver ICs with on-chip filtering," *ISSCC Dig. Tech. Pap.,* pp. 148-149, Feb. 1995.

[8] T. Yamawaki, M. Kokubo, K. Irie, H. Matsui, K. Hori, T. Endou, H. Hagisawa, T. Furuya, Y. Shimizu, M. Katagishi, J.R. Hildersley, "A 2.7-V GSM RF transceiver IC," *IEEE J. Solid-State Circuits,* vol. 32, pp. 2089-2096, Dec. 1997.

[9] J.C. Rudell, J.-J. Ou, T.B. Cho, G. Chien, F. Brianti, J.A. Weldon, and P.R. Gray, "A 1.9-GHz wide-band IF double conversion CMOS receiver for cordless telephone applications ," *IEEE J. Solid-State Circuits,* vol.32, pp. 2071-2088, Dec. 1997.

[10] J.C. Rudell, J.-J. Ou, T.B. Cho, G. Chien, F. Brianti, J.A. Weldon, and P.R. Gray, "A 70-mW seventh-order filter with 7-50 MHz cutoff frequency and programmable boost and group delay equalization," *IEEE J. Solid-State Circuits,* vol.32, pp. 1987-1999, Dec. 1997.

[11] C. Yoo, S.-W. Lee, and W. Kim, "A ±1.5-V, 4-MHz CMOS continuous-time filter with a single-integrator based tuning," *IEEE J. Solid-State Circuits,* vol.33, pp. 18-27, Jan. 1998.

[12] J.K. Pyykonen, "A low distortion wideband active-RC filter for a multicarrier base station transmitter," *Proc. IEEE Int. Symp. Circuits Syst.,* vol. 1, pp. 244-247, May 2001.

[13] A.M. Durham, W. Redman-White, and J.B. Hughes, "High-linearity continuous-time filter in 5V VLSI CMOS," *IEEE J. Solid-State Circuits,* vol. 27, pp. 1270-1276, Sept. 1992.

[14] J. van der Plas, "MOSFET-C filter, with low excess noise and accurate automatic tuning," *IEEE J. Solid-State Circuits,* vol. 26, pp. 922-929, July 1991.

[15] Y. Tsividis, M. Banu, and J. Khoury, "Continuous-time MOSFET-C filters in VLSI," *IEEE J. Solid-State Circuits,* vol. 21, pp. 15-30, Feb. 1986.

[16] R.P. Sallen and E.L. Key, "A practical method of designing RC active filters," *IRE Trans. Circuit Theory,* pp. 74-85, Mar. 1955.

[17] J. Tow, "A step-by-step active filter design," *IEEE Spectrum,* pp. 64-68, Dec. 1969.

[18] A.M. Durham, and W. Redman-White, "Integrated continuous-time balanced filters for 16-b DSP interfaces" *IEEE J. Solid-State Circuits,* vol. 28, pp. 835-839, July 1993.

[19] S. Signell and K. Mossberg, "Design of high-frequency active-RC filters using passive compensation methods," *Proc. IEEE Int. Symp. Circuits Syst.,* vol. 3, pp. 2669-2672, June 1988.

[20] K. Martin and D.A. Johns, *Analog Integrated Circuit Design,* John Wiley and Sons, New York, 1997.

[21] H. Ng, R. Ziazedh, and D. Allstot, "A multistage amplifier technique with embedded frequency compensation," *IEEE J. Solid-State Circuits,* vol.34, pp. 339-347, Mar. 1999.

[22] G. Palumbo and S. Pennisi, "Design methodology and advances in nested-Miller copensation," *IEEE Trans. Circuits Syst. I,* vol. 49, pp. 893-903, July 2002.

[23] G. Palmisano and G. Palumbo, "An optimized compensation strategy for two-stage CMOS opamps," *IEEE Trans. Circuits Syst. I,* vol. 42, pp. 178-182, Mar. 1995.

[24] B. Razavi, *Design of Analog CMOS Integrated Circuits,* McGraw-Hill, New York, 2001.

[25] F. Yang and C. Enz, "A low-distortion BiCMOS seventh-order Bessel filter operating at 2.5 V supply," *IEEE J. Solid-State Circuits,* vol. 31, pp. 321-330, Mar. 1996.

[26] J. Egerer, T. Desel, Popken, J. Boxho, D. Macq, and J.-P. Cornil, "A low distortion linear-tunable continuous-time 488 kHz fifth-order Bessel filter ," *Proc. IEEE Int. ASIC Conf.,* pp. 47-50, Sept. 1998.

[27] H. Huang and E.K.F. Lee, "Design of low-voltage CMOS continuous-time filter with on-chip automatic tuning," *IEEE J. Solid-State Circuits,* vol. 36, pp. 1168-1177, Aug. 2001.

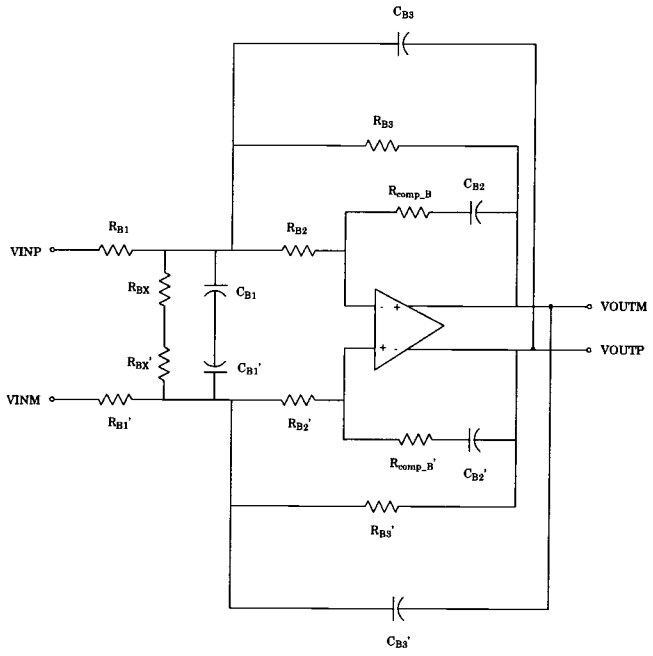
[28] S.D. Willingham, K.W. Martin, and A. Ganesan, "A BiCMOS low-distortion 8-MHz low-pass filter," *IEEE J. Solid-State Circuits,* vol. 28, pp. 1234-1245, Dec. 1993.

[29] V. Gopinathan, M. Tarsia, and D. Choi, "Design considerations and implementation of a programmable high-frequency continuous-time filter and variable-gain amplifier in submicrometer CMOS ," *IEEE J. Solid-State Circuits,* vol. 34, pp. 1698-1707, Dec. 1999.

APPENDICES

APPENDIX A. Biquad Transfer Functions

First Stage Biquad:



$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{K\omega_o^2}{s^2 + \frac{\omega_o}{Q} + \omega_o^2} \tag{A1}$$

where,

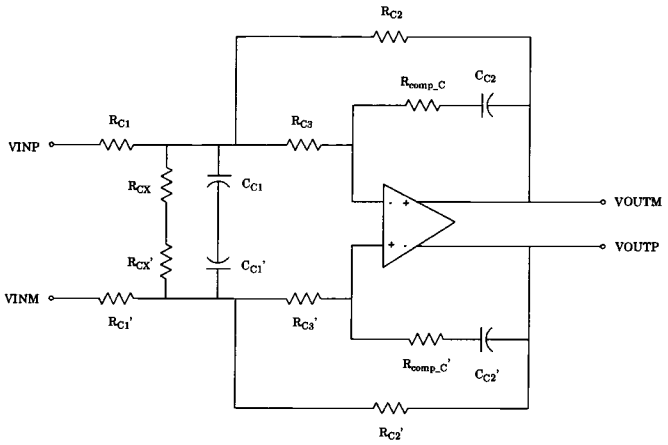
$$K = \frac{R_{B3}}{R_{B1}} \tag{A2}$$

$$\frac{\omega_o}{Q} = \frac{1}{(C_{B1} + C_{B3})}\left(\frac{1}{R_{B1}} + \frac{1 - \frac{C_{B3}}{C_{B2}}}{R_{B2i}} + \frac{1}{R_{B3}} + \frac{1}{R_{BX}}\right) \tag{A3}$$

$$\omega_o^2 = \frac{1}{(C_{B1} + C_{B3})C_{B2}R_{B2i}R_{B3}} \tag{A4}$$

$$R_{B2} = R_{B2i} - R_{comp\_B} \tag{A5}$$

Second Stage Biquad :



$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{K\omega_o^2}{s^2 + \frac{\omega_o}{Q} + \omega_o^2} \tag{A6}$$

where,

$$K = \frac{R_{C2}}{R_{C1}} \tag{A7}$$

$$\frac{\omega_o}{Q} = \frac{1}{C_{C5}}\left(\frac{1}{R_{C1}} + \frac{1}{R_{C2}} + \frac{1}{R_{C3i}} + \frac{1}{R_{CX}}\right) \tag{A8}$$

$$\omega_o^2 = \frac{1}{R_{C2}R_{C3i}C_{C5}C_{C6}} \tag{A9}$$

$$R_{C3} = R_{C3i} - R_{comp\_C} \tag{A10}$$

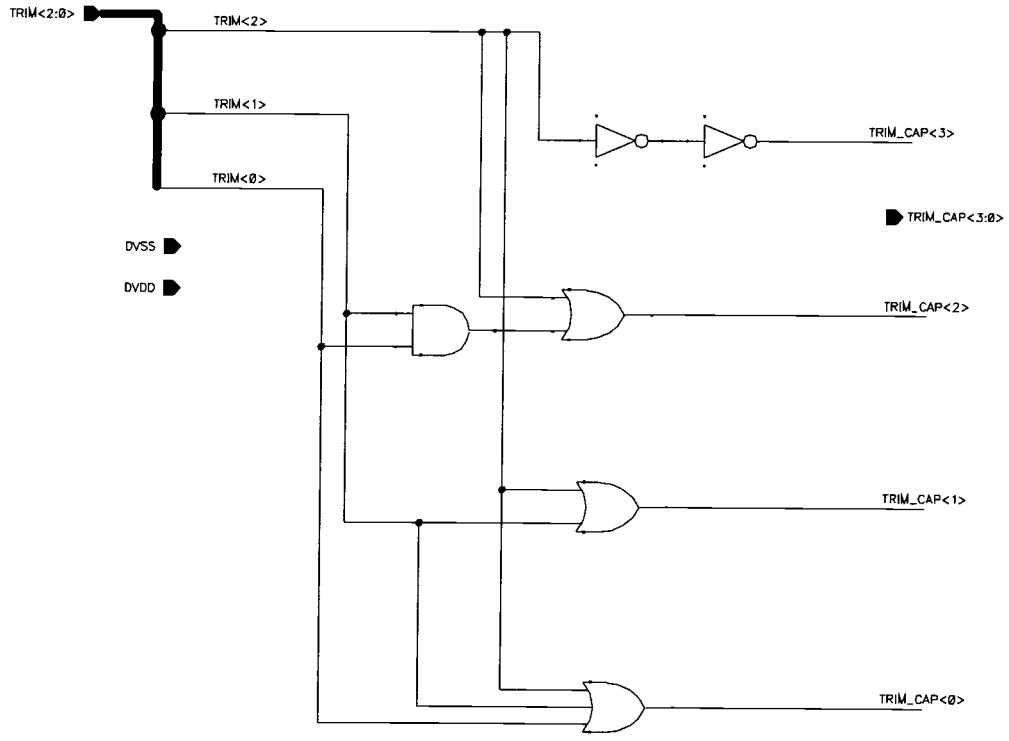
APPENDIX B. Comparison between three R-C internal feedback compensation schemes (two with ideal resistors and one with a MOS resistor)

Scheme	Corner	BW	PM	%ΔBW	%ΔPM	THD	Capacitors
Scheme2	Weak			-34%	-64%		
(with ideal	Nominal	645MHz	70^o	-	-	88dB	1.6pF, 1pF
resistors)	Strong			higher%	higher%		
Scheme3	Weak			-40%	-40%		
(with ideal	Nominal	556MHz	73^o	-	-	80dB	2pF, 1pF
resistors)	Strong			higher%	higher%		
Scheme3	Weak			-22%	0%		
(with MOS	Nominal	583MHz	70^o	-	-	80dB	2pF, 1.5pF
resistors)	Strong			+37%	+1%		

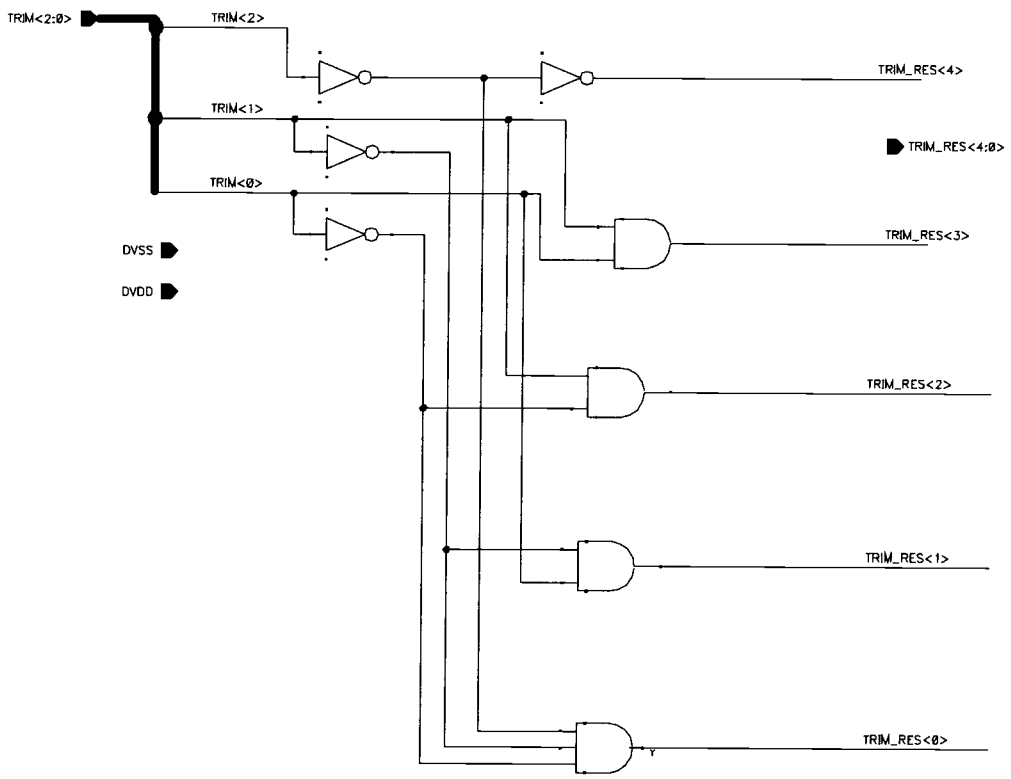
Note: The weak corner refers to a weak process at 125°C, the nominal corner refers to a nominal process at 27°C and the strong corner refers to a strong process at -40^oC.

APPENDIX C. Diagrams of the Digital Blocks

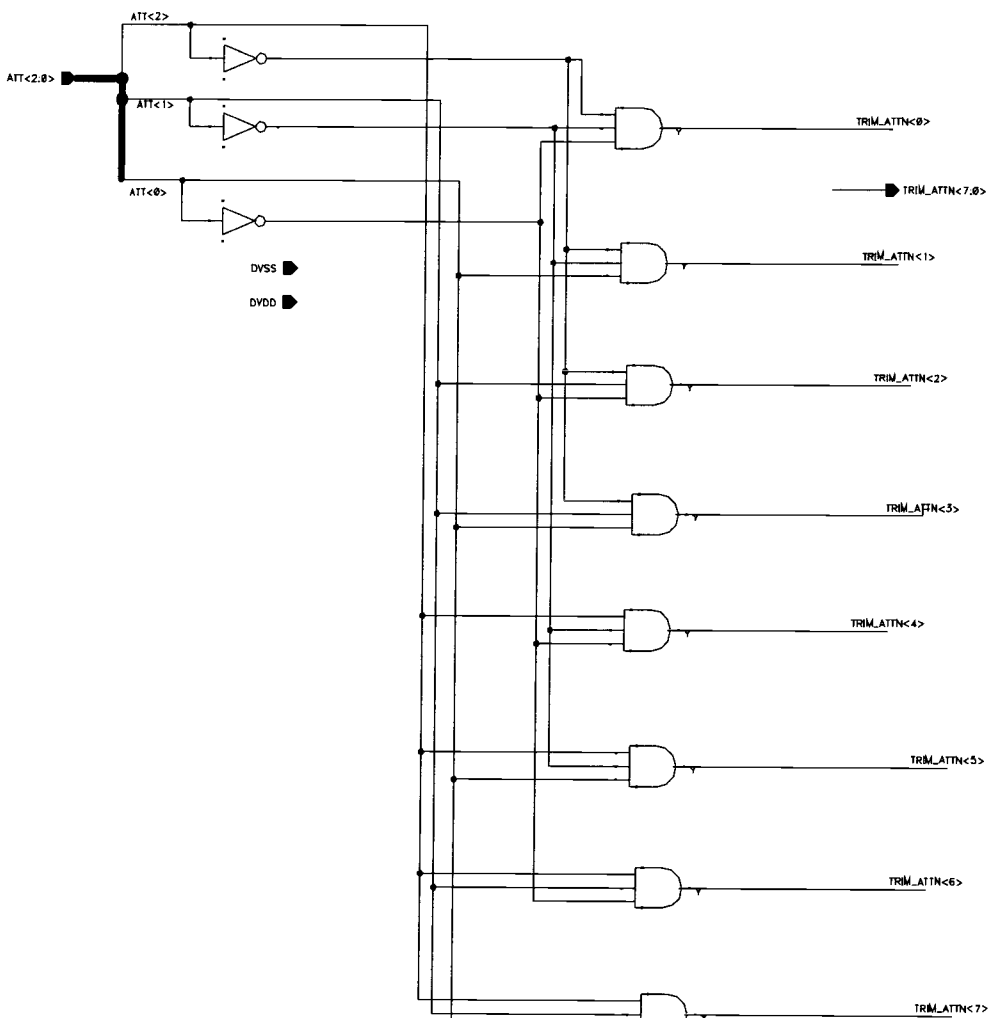
Logic generation for frequency trimming of capacitors :



Logic generation for frequency trimming of resistors :

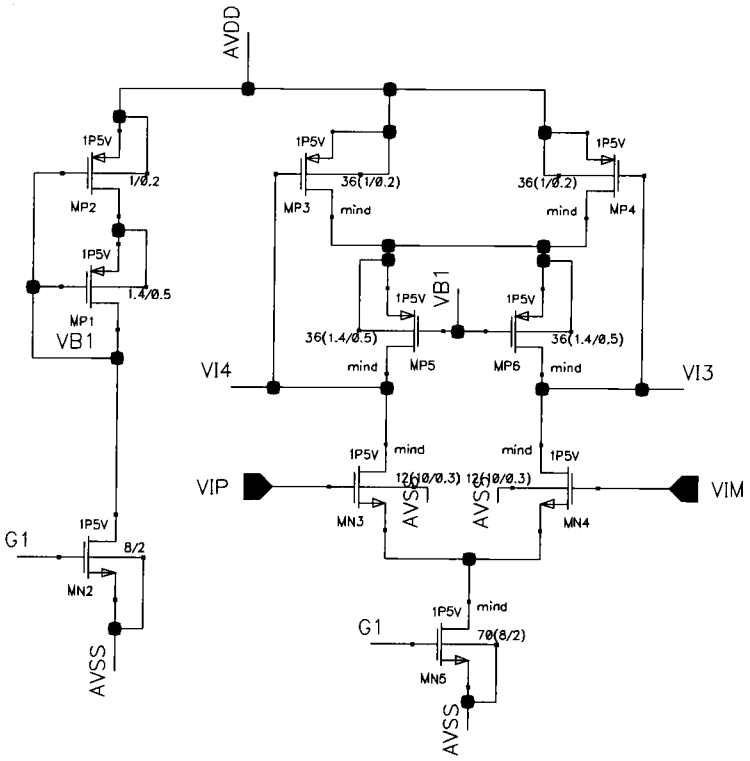


Logic generation for attenuation trimming of resistors :

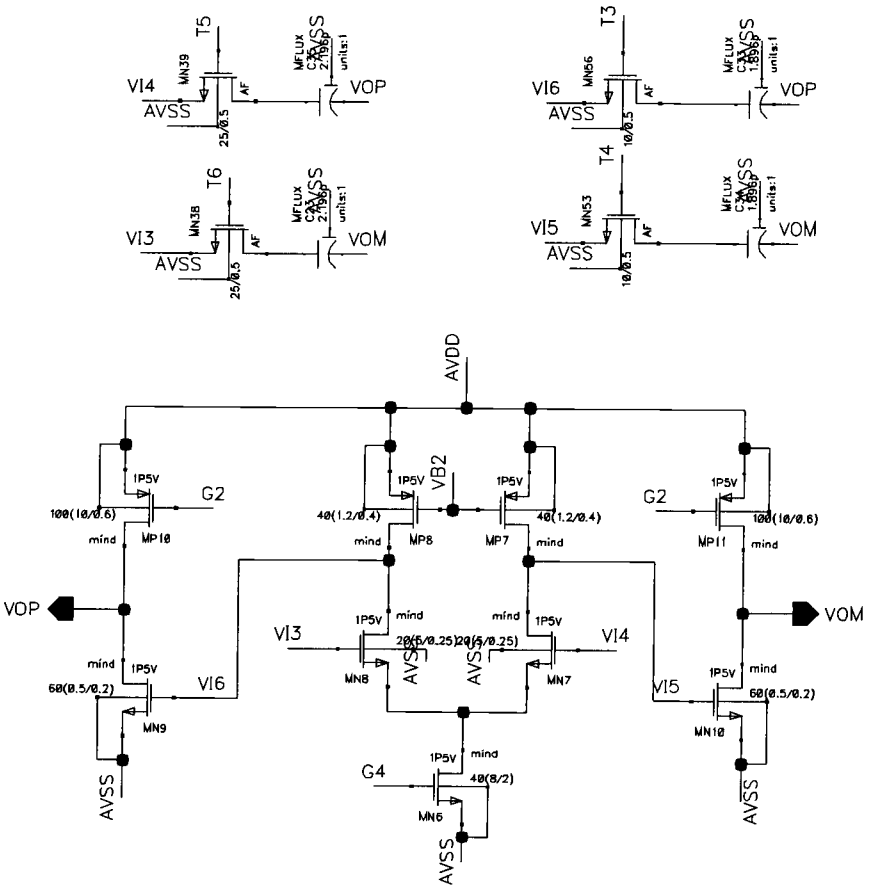


APPENDIX D. Circuit Diagrams from Cadence

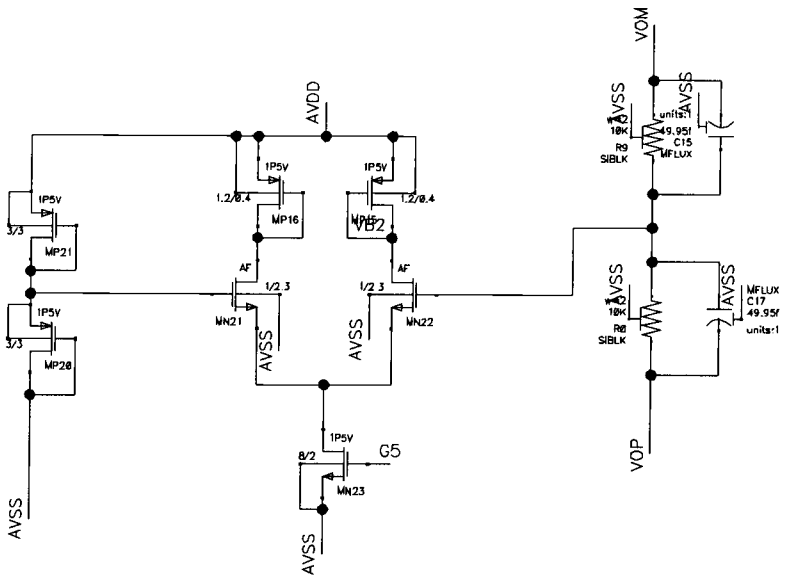
First stage of opamp :



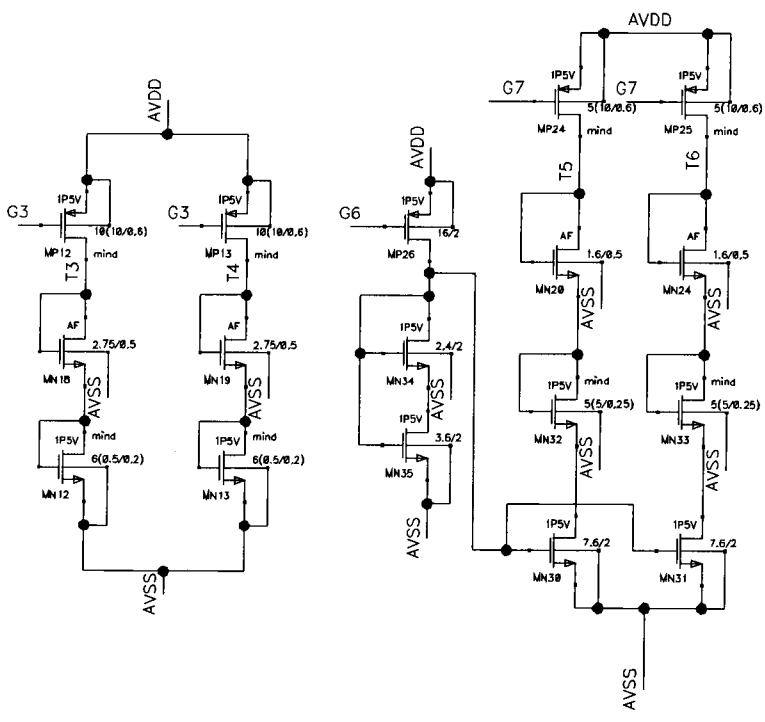
Second and third stages of opamp and nested miller compensation loops :



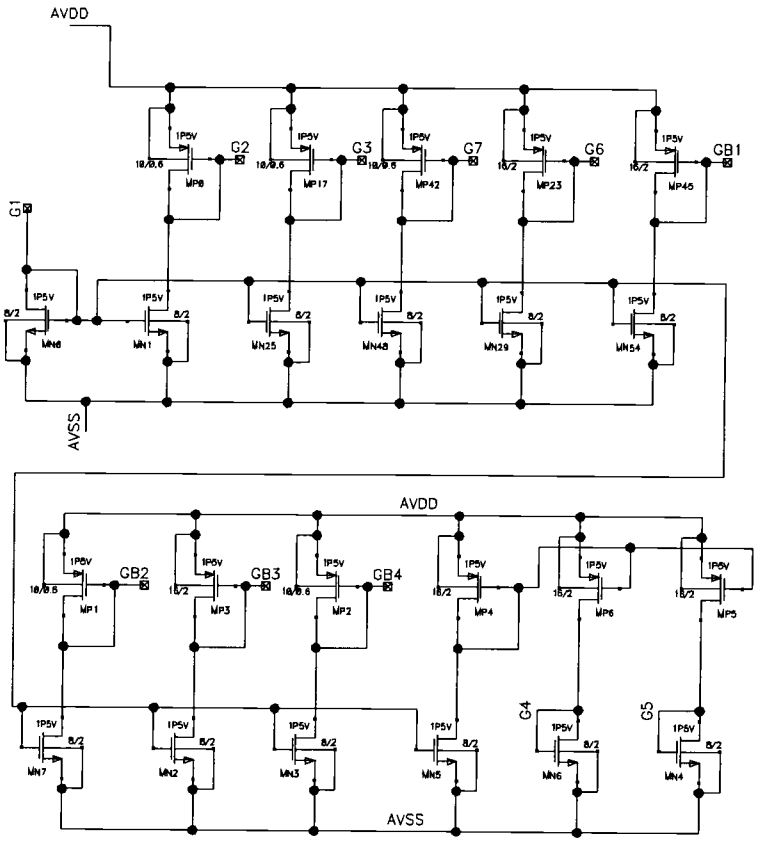
Common-mode Feedback for second and third stages :



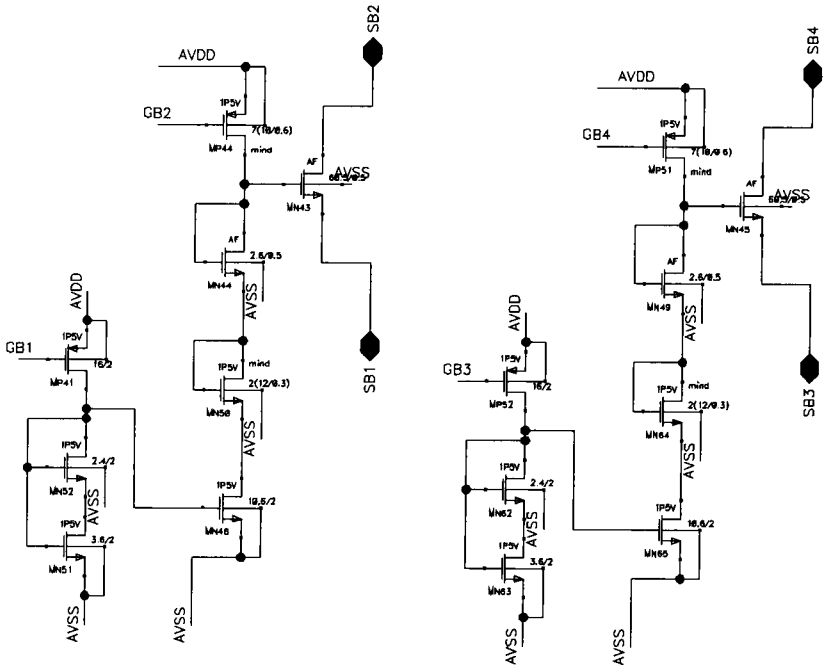
Biasing for compensation MOS transitors :



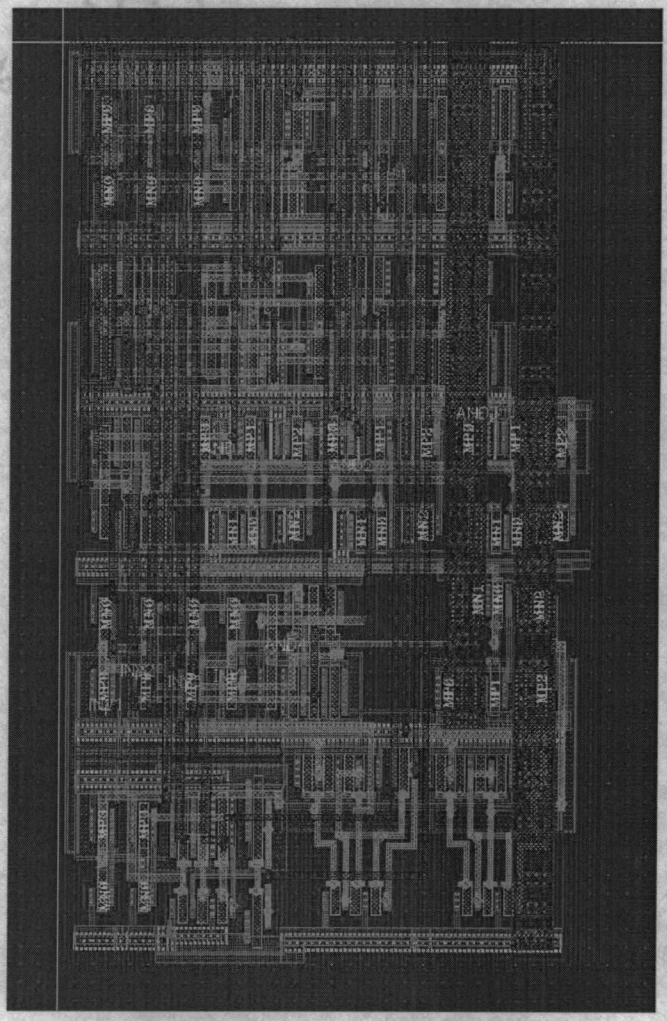
Main biasing for opamp :



MOS resistors in integrator feedback loop with biasing circuitry :



APPENDIX E. Layout of the Digital Logic Block



APPENDIX F. MATLAB Code for Generating Element Values

```
%%%%%%%%%%%%%%%%%
clear;
close all;
%format long e;

fid = fopen('out1.txt','w');

% coefficients from transfer function
b11 = 5.953e07 ; % 1st stage
b22 = 6.049e15 ;
c11 = 1.437e08 ; % 2nd stage
c22 = 6.049e15 ;

% 1st Stage (b)(high Q SAB)
cb1 = 6e-12 ;
cb3 = 6e-12 ;
rb1 = 4.5e3 ;
rbx = 1e3 ;

% 2nd Stage (c)(low Q SAB)
cc5 = 25e-12 ;
rcx = 1e3 ;
gcx = 1/rcx ;
```

```
gcx(1) = gcx;
rcx(1) = rcx;
ugbw = 500e6 ;% unity gain BW of opamp

% gains of the stages for node-scaling in dB
gb1_db = -1.483 ;
gb2_db(1) = 1.483;
gb = 10^(gb1_db/20);
gc(1) = 10^(gb2_db(1)/20);
wb = b22^0.5;
wc = c22^0.5;
qc = wc/c11 ;
qb = wb/b11;

% 1st stage
%disp('1st Stage values');

rb3 = rb1*gb ;
rb2 =
1/(((cb1+cb3)*((1/(wb*qb))+(rb3*cb3))*wb*wb)-((1/rb1)+(1/rb3)+(1/rbx)));
cb2 = 1/(rb2*rb3*wb*wb*(cb1+cb3));
rcomp_b = 1/(2*pi*ugbw*cb2) ;
rb2_new = rb2-rcomp_b;

% 2nd stage
```

```
%disp('2nd Stage values');

gc_temp = gc(1);
m = 0.98*((cc5*wc-qc*gcx)^2)/(4*qc*qc*(1+gc_temp)*wc*wc*cc5*cc5);
m1 = m;
cc6 = m*cc5 ;
temp1 = (cc5*wc)-(qc*gcx);
temp2 = 4*qc*qc*wc*wc*(1+gc(1))*m*cc5*cc5;
temp3 = sqrt((temp1^2)-temp2);
temp4 = 2*qc*wc*wc*m*cc5*cc5;
rc2(1) = (temp1-temp3)/temp4;
rc1(1) = rc2(1)/gc(1);
rc3 = 1/(wc*wc*cc5*cc5*m*rc2(1));
rcomp_c = 1/(2*pi*ugbw*cc6);
rc3_new(1) = rc3 - rcomp_c ;

%gc_tot(1) = 1/rc1+1/rc2+1/rc3+1/rcx;
%cc6_new(1) = cc6-(1/(rc3*2*pi*ugbw))+(1/((rc3^2)*2*pi*ugbw*gc_tot));
%cc5_new(1) = 0.5*cc5*cc6/(cc6_new+(1/(2*pi*ugbw*rc3)));
m=m1;

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

for i=2:8,
```

```
gb2_db(i) = 1.483 - 3*(i-1);
gc(i) = 10^(gb2_db(i)/20);
gc_temp=gc(i);
gcx_temp = solve('m =
0.98*((cc5*wc-
qc*gcx_temp)^2)/(4*qc*qc*(1+gc_temp)*wc*wc*cc5*cc5)','gcx_temp');
gcx(i) = eval(gcx_temp(2));
rcx(i) = 1/gcx(i);
temp1 = (cc5*wc)-(qc*gcx(i));
temp2 = 4*qc*qc*wc*wc*(1+gc(i))*m*cc5*cc5;
temp3 = sqrt((temp1^2)-temp2);
temp4 = 2*qc*wc*wc*m*cc5*cc5;
rc2(i) = (temp1-temp3)/temp4;
rc1(i) = rc2(i)/gc(i);
rc3 = 1/(wc*wc*cc5*cc5*m*rc2(i));
rc3_new(i) = rc3 - rcomp_c;
%gc_tot(i) = 1/rc1(i)+1/rc2(i)+1/rc3(i)+1/rcx(i);
%cc6_new(i) =
cc6-(1/((rc3(i))*2*pi*ugbw))+(1/(((rc3(i))^2)*2*pi*ugbw*gc_tot(i)));
%cc5_new(i) = 0.5*cc5*cc6/(cc6_new(i)+(1/(2*pi*ugbw*rc3(i))));

end

%% Trimming of 1st stage
```

```
rm1=rb1; % required input res
rn1=rb3;% required feedback res
rb2 = rb2; % res to opamp input

r1=rm1/(2*1.4); % lower tee res (not the one to gnd !)
r2=rn1/(2*1.4); % upper tee res (not the one to gnd !)
rbx=1e3;


xa=25; % assumed percentage variation
xb=35; % assumed percentage variation
y=10; % corrected percentage variation

x1a=1-(xa/100); % 0.75
x1b=1-(xb/100); % 0.65
x2a=1+(xa/100); % 1.25
x2b=1+(xb/100); % 1.35
y1=1-(y/100); % 0.9
y2=1+(y/100); % 1.1

%ra2 = ra1;
%rc2 = rc1;

% nom case
%disp('nominal case');
```

```
rp = r1;
rna =(rp*rp)/(rm1-(rp+rp));
rnm = ((rp*rp)+(rp*rna)+(rp*rna))/rp;

rq = r2;
rnb =(rq*rq)/(rn1-(rq+rq));
rnn = ((rq*rq)+(rq*rnb)+(rq*rnb))/rq;

rnx=1/((1/rbx)-(1/rnm)-(1/rnn));

Rb3_nom = rnb; % upper tee res to gnd
Rb1_nom = rna; % lower tee res to gnd
Rbx_nom = rnx ;

%low case
%disp('low case a');

rp = x1a*r1;
rla =(rp*rp)/((y1*rm1)-(rp+rp));
rlm = ((rp*rp)+(rp*rla)+(rp*rla))/rp;

rq = x1a*r2;
rlb =(rq*rq)/((y1*rn1)-(rq+rq));
rln = ((rq*rq)+(rq*rlb)+(rq*rlb))/rq;
```

```
rlx=1/((1/(y1*rbx))-(1/rlm)-(1/rln));

Rb3_low_a = rlb/x1a;
Rb1_low_a = rla/x1a;
Rbx_low_a = rlx/x1a;

%disp('low case b');

rp = x1b*r1;
rla =(rp*rp)/((y1*rm1)-(rp+rp));
rlm = ((rp*rp)+(rp*rla)+(rp*rla))/rp;

rq = x1b*r2;
rlb =(rq*rq)/((y1*rn1)-(rq+rq));
rln = ((rq*rq)+(rq*rlb)+(rq*rlb))/rq;

rlx=1/((1/(y1*rbx))-(1/rlm)-(1/rln));

Rb3_low_b = rlb/x1b;
Rb1_low_b = rla/x1b;
Rbx_low_b = rlx/x1b;

%high case
%disp('high case a');
```

```
rp = x2a*r1;
rha =(rp*rp)/((y2*rm1)-(rp+rp));
rhm = ((rp*rp)+(rp*rha)+(rp*rha))/rp;

rq = x2a*r2;
rhb =(rq*rq)/((y2*rn1)-(rq+rq));
rhn = ((rq*rq)+(rq*rhb)+(rq*rhb))/rq;

rhx=1/((1/(y2*rbx))-(1/rhm)-(1/rhn));

Rb3_high_a = rhb/x2a;
Rb1_high_a = rha/x2a;
Rbx_high_a = rhx/x2a;

%disp('high case b');

rp = x2b*r1;
rha =(rp*rp)/((y2*rm1)-(rp+rp));
rhm = ((rp*rp)+(rp*rha)+(rp*rha))/rp;

rq = x2b*r2;
rhb =(rq*rq)/((y2*rn1)-(rq+rq));
rhn = ((rq*rq)+(rq*rhb)+(rq*rhb))/rq;
```

```
rhx=1/((1/(y2*rbx))-(1/rhm)-(1/rhn));

Rb3_high_b = rhb/x2b;
Rb1_high_b = rha/x2b;
Rbx_high_b = rhx/x2b;

%disp('rc1 trim res values'); % lower tee res to gnd

%rc1_nom = rna;
%rc1_low = rla/x1;
%rc1_high = rha/x2;

%disp('Rb3 trim res values'); % upper tee res to gnd
%Rb3_nom = rnb;
%Rb3_low = rlb/x1;
%Rb3_high = rhb/x2;

%disp('Rbx trim res values');

%Rbx_nom = rnx;
%Rbx_low = rlx/x1;
%Rbx_high = rhx/x2;

%r1=rlm/x1;
%r2=rln/x1;
```

```
%% R_par= 1/((1/r1)+(1/r2)+(1/Rbx_low))

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

%disp('Actual resistor values in resistor array');

%disp('Rb1 actual values');% low to high

R1 = Rb1_low_b;
R2 = Rb1_low_a - Rb1_low_b;
R3 = Rb1_nom - Rb1_low_a;
R4 = Rb1_high_a - Rb1_nom;
R5 = Rb1_high_b - Rb1_high_a;
a(1)=R1;
a(2)=R2;
a(3)=R3;
a(4)=R4;
a(5)=R5;

fprintf(fid,'\n *************** 1st Stage values ****************** \n\n');
fprintf(fid,' Cb1 = %d \n\n Cb2 = %d \n\n Cb3 = %d \n\n',cb1,cb2,cb3);
fprintf(fid,'Rb1 upper value = %d \n\n',rm1/(2*1.4));
fprintf(fid,'Rb1 tee actual values : \n\n R1 = %d R2 = %d R3 = %d R4 = %d R5
= %d\n\n\n', R1,R2,R3,R4,R5);
```

```
%disp('Rb3 actual values');% low to high for increasing R

R1 = Rb3_low_b;
R2 = Rb3_low_a - Rb3_low_b;
R3 = Rb3_nom - Rb3_low_a;
R4 = Rb3_high_a - Rb3_nom;
R5 = Rb3_high_b - Rb3_high_a;
a(6)=R1;
a(7)=R2;
a(8)=R3;
a(9)=R4;
a(10)=R5;

        fprintf(fid,'Rb3 upper value = %d \n\n',rn1/(2*1.4));
fprintf(fid,'Rb3 tee actual values : \n\n R1 = %d R2 = %d R3 = %d R4 = %d R5
= %d\n\n\n', R1,R2,R3,R4,R5);

%disp('Rbx actual values'); % high to low

R1 = Rbx_high_b;
R2 = Rbx_high_a - Rbx_high_b;
R3 = Rbx_nom - Rbx_high_a;
R4 = Rbx_low_a - Rbx_nom;
R5 = Rbx_low_b - Rbx_low_a;
```

```
a(11)=R1;
a(12)=R2;
a(13)=R3;
a(14)=R4;
a(15)=R5;

fprintf(fid,'Rbx actual values : \n\n R1 = %d R2 = %d R3 = %d R4 = %d R5 =
%d\n\n\n', R1,R2,R3,R4,R5);

%disp('Rb2 actual values'); % high to low
{R1,R2,R3,R4,R5}=
solve('x2b*R1=y2*rb2_new','x2a*(R1+R2)=y2*rb2_new','R1+R2+R3=rb2_new',
'x1a*(R1+R2+R3+R4)=y1*rb2_new','x1b*(R1+R2+R3+R4+R5)=y1*rb2_new',
'R1,R2,R3,R4,R5');
R1=eval(R1);
R2=eval(R2);
R3=eval(R3);
R4=eval(R4);
R5=eval(R5);
a(16)=R1;
a(17)=R2;
a(18)=R3;
a(19)=R4;
a(20)=R5;
```

```
fprintf(fid,'Rb2 actual values : \n\n R1 = %d R2 = %d R3 = %d R4 = %d R5 =
%d\n\n\n', R1,R2,R3,R4,R5);

%%%%%%%%% Trimming of 2nd stage

fprintf(fid,'*************** 2nd Stage values *************** \n\n');

% trimming capacitors

%disp('cc5 trim values');

k=cc5;
{c1,c2,c3,c4,c5}=solve('x2b*c1=y2*k','x2a*(c1+c2)=y2*k','c1+c2+c3=k',
'x1a*(c1+c2+c3+c4)=y1*k','x1b*(c1+c2+c3+c4+c5)=y1*k','c1,c2,c3,c4,c5');
c1=eval(c1);
c2=eval(c2);
c3=eval(c3);
c4=eval(c4);
c5=eval(c5);
a(21)=c1;
a(22)=c2;
a(23)=c3;
a(24)=c4;
a(25)=c5;
```

```
fprintf(fid,'Cc5 actual values : \n\n C1 = %d C2 = %d C3 = %d C4 = %d C5
= %d\n\n\n', c1,c2,c3,c4,c5);

% a= 25% b=35% l=lower side h=higher side

%disp('cc6 trim values');

k= cc6;
{c1,c2,c3,c4,c5}= solve('x2b*c1=y2*k',
'x2a*(c1+c2)=y2*k','c1+c2+c3=k','x1a*(c1+c2+c3+c4)=y1*k',
'x1b*(c1+c2+c3+c4+c5)=y1*k','c1,c2,c3,c4,c5');

c1=eval(c1);
c2=eval(c2);
c3=eval(c3);
c4=eval(c4);
c5=eval(c5);
a(26)=c1;
a(27)=c2;
a(28)=c3;
a(29)=c4;
a(30)=c5;

fprintf(fid,'Cc6 actual values : \n\n C1 = %d C2 = %d C3 = %d C4 = %d C5
= %d\n\n\n', c1,c2,c3,c4,c5);
```

```
rm1=rc1(1); % required input res
rn1=rc2(1); % required feedback res

r1=rm1/(2*1.15); % lower tee res (not the one to gnd !)
r2=rn1/(2*1.45); % upper tee res (not the one to gnd !)

rc1_u = r1;
rc2_u = r2;

for i=1:8,

rm1=rc1(i); % required input res
rn1=rc2(i); % required feedback res

rp = r1;
rna =(rp*rp)/(rm1-(rp+rp));
rnm = ((rp*rp)+(rp*rna)+(rp*rna))/rp;

rq = r2;
rnb =(rq*rq)/(rn1-(rq+rq));
rnn = ((rq*rq)+(rq*rnb)+(rq*rnb))/rq;

rnx=1/((1/rcx(i))-(1/rnm)-(1/rnn));
```

```
rc1_t(i) = rna; % lower tee res to gnd
rc2_t(i) = rnb; % upper tee res to gnd
rcx_new(i) = rnx;

end

rc1_t_act(8) = rc1_t(8);

for i=7:-1:1,
rc1_t_act(i)= rc1_t(i) - rc1_t(i+1);
end

rc2_t_act(1) = rc2_t(1);
for i=2:8,
rc2_t_act(i)= rc2_t(i) - rc2_t(i-1);
end

rc3_new_act(1) = rc3_new(1);
for i=2:8,
rc3_new_act(i)= rc3_new(i) - rc3_new(i-1);
end

rcx_new_act(1) = rcx_new(1);
for i=2:8,
rcx_new_act(i)= rcx_new(i) - rcx_new(i-1);
```

```
end

fprintf(fid,'Rc1_u = %d ',rc1_u);
b(1)=rc1_u;

fprintf(fid,'\n\n');

for i=1:4,
fprintf(fid,'Rc1_t_act(%d)= %d ',i,rc1_t_act(i)); % decreasing R with decreasing gain
b(i+1)=rc1_t_act(i);
end

fprintf(fid,'\n');

for i=5:8,
fprintf(fid,'Rc1_t_act(%d)= %d ',i,rc1_t_act(i));
b(i+1)=rc1_t_act(i);
end

fprintf(fid,'\n\n');

fprintf(fid,'Rc2_u = %d ',rc2_u);
b(10)=rc2_u;
fprintf(fid,'\n\n');
```

```
for i=1:4,
fprintf(fid,'Rc2_t_act(%d)= %d ',i,rc2_t_act(i));% increasing R for decreasing gain
b(i+10)=rc2_t_act(i);
end

fprintf(fid,'\n');

for i=5:8,
fprintf(fid,'Rc2_t_act(%d)= %d ',i,rc2_t_act(i));
b(i+10)=rc2_t_act(i);
end

fprintf(fid,'\n\n');

for i=1:4,
fprintf(fid,'Rc3_act(%d)= %d ',i,rc3_new_act(i)); %increasing R for decreasing gain
b(i+18)=rc3_new_act(i);
end

fprintf(fid,'\n');

for i=5:8,
fprintf(fid,'Rc3_act(%d)= %d ',i,rc3_new_act(i));
b(i+18)=rc3_new_act(i);
end
```

```
fprintf(fid,'\n\n');

for i=1:4,
fprintf(fid,'Rcx_act(%d)= %d ',i,rcx_new_act(i));%increasing R for decreasing gain
b(i+26)=rcx_new(i);
end

fprintf(fid,'\n');

for i=5:8,
fprintf(fid,'Rcx_act(%d)= %d ',i,rcx_new_act(i));
b(i+26)=rcx_new(i);
end

fprintf(fid,'\n\n');

flag = 0;

for i = 1:30,
if a(i)¡0
flag = 1;
end

end
```

```
for i = 1:34,
if b(i)¡0
flag = 1;
end

end

if flag == 0
fprintf(fid,'\n\n\t\t\t\tSUCCESSFUL !! YOO HOO !! \n\n');
else
fprintf(fid,'\n\n\t\t\t\tUNSUCCESSFUL .... TRY AGAIN \n\n');
end

fclose(fid);
```