Automatic RFIC Synthesis

Automation for Passive Mixer Design

A Project Report

submitted by

NARASIMHAN SRIKANTH

in partial fulfilment of requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
May 2024

Contents

Chapter 1

Introduction

1.1 Overview

Design of Integrated Circuits involves tedious computations and multiple simulations. To meet the user's specifications, a designer has to tune the circuit variables through repeated simulations. While designing any circuit, the most common first step is to arrive at a set of values for the circuit components based on theoretical equations and hand calculations. These values must be fine-tuned using a circuit simulator, incorporating more accurate component models. These two steps usually take significant time and effort in the design procedure.

Artificial Intelligence and Machine Learning algorithms have become increasingly popular solutions for automating manual tasks in engineering. Designers have begun adopting AI and ML techniques in VLSI systems to solve optimisation problems such as power and area.

Automatic RFIC synthesis aims to reduce the effort by automating the schematic design procedure. The automation routine will provide a set of component values that satisfy the user's specifications for the circuit. In this project, I have automated the schematic design of **Voltage mode and Current mode Passive Downconversion mixers**. **Gradient Descent** algorithm was used to achieve the best netlist for the given specifications.

Passive mixers offer good impedance matching over a wide range of frequencies and better linearity than traditional active mixers. The transistors in the passive mixer operate as switches that allow the impedance presented at the baseband port to pass through to the RF port. This transparency has led to the designing and implementing of wideband tunable-impedance match Passive mixer-first receivers.

1.1.1 The Passive Downconversion Mixer

A mixer is an essential component in the receiver because it retrieves the baseband message signal from the RF carrier. In the receiver path, the downconversion mixer senses the RF signal at the input "RF port" and the Local Oscillator (LO) signal at its "LO port". The output is called the "baseband port". This mixer performs frequency translation from RF to baseband by multiplying the LO waveform with the input RF waveform. Figure 1.1 illustrates the role of a downconversion mixer.

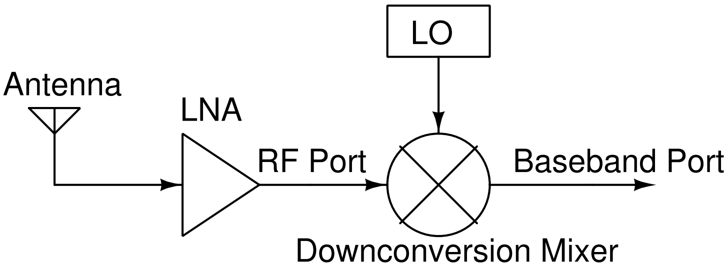


Figure 1.1: Receiver path diagram

The Local Oscillator (referred to as LO from here on) waveform in this project is always assumed to be close to an abruptly switching square wave. A small rise-time and fall-time is considered because abrupt switching is not achievable in practice. The mixing operation can be viewed as multiplying the square wave with the incoming RF waveform.

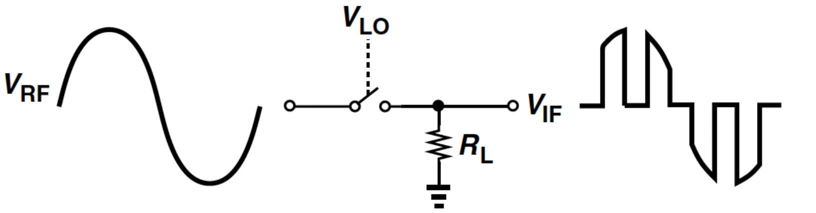
A passive downconversion mixer is one where the transistors do not operate as amplifying devices. The transistors behave like switches in this type of mixer. This means the transistors operate either in the cut-off region when switched off or in the linear (triode) region when switched on. Figure 1.2 shows the frequency translation from RF to baseband using switching.

1.1.2 Types of Passive Mixers

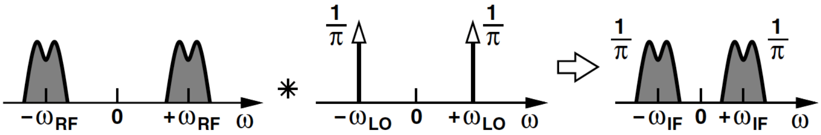
Based on the type of RF input, there are 2 types of passive mixers: the voltage mode passive mixer and the current mode passive mixer. The difference in operation between these two devices lies in the RF input. As the name suggests, the input to the voltage mode mixer is an RF voltage and the input to the current mode mixer is an RF current.

The passive mixer is called a return-to-zero mixer when the load presented at the output is a resistance. On the contrary, when the load is purely capacitive, the mixer is called a non-return-to-zero mixer (also known as the sampling mixer).

Passive mixers can be single-balanced or double-balanced. In the single-



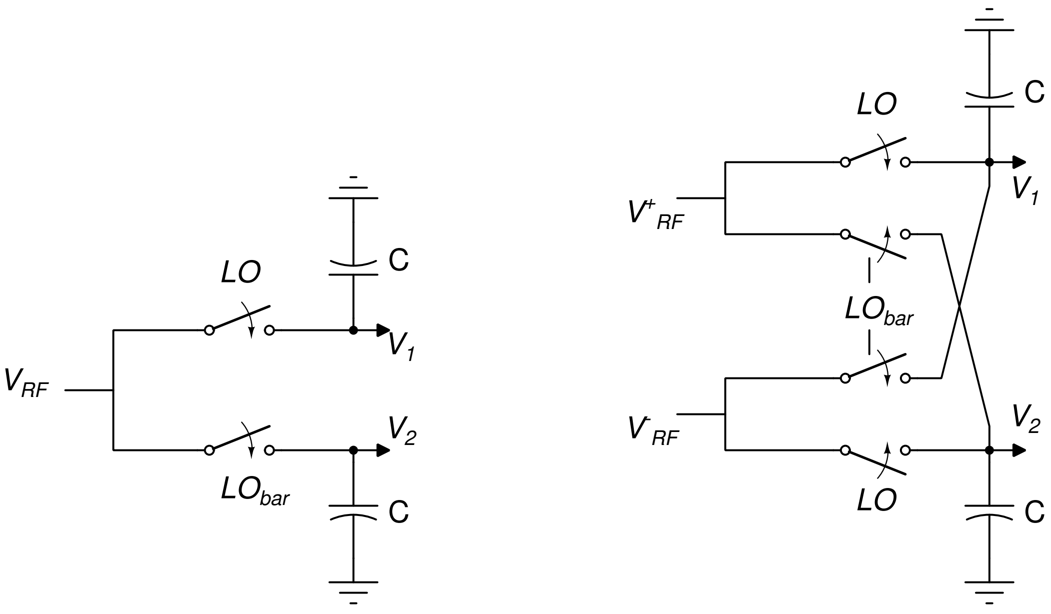
(a) switch connected to a simple resistive load



(b) corresponding spectra

Figure 1.2: frequency translation in a passive mixer using a switch
ref: RF Microelectronics, *Razavi*

balanced passive mixer, the RF input is single-ended and the baseband output is differential. In the double-balanced passive mixer, the RF input is balanced and the output baseband is differential. Figure 1.3 shows single and double-balanced versions of the passive sampling mixer.



(a) single-balanced mixer

(b) double-balanced mixer

Figure 1.3: sampling mixer

Chapter 2

Optimisation Algorithm

The **Gradient Descent** algorithm has been used in this project. The key aspects of the algorithm are the choice of the Loss function and the computation of the new set of circuit parameters in every iteration.

2.1 Choice of Loss Function

The specifications that I have chosen to optimise for the passive downconversion mixer (current and voltage mode) are the input impedance matching looking in from the input port (S_{11}), the voltage gain, the integrated single sideband (SSB) noise figure (integrated over the bandwidth of operation), the third order input intercept point (IIP_3) and power consumption. The right cost function must be chosen to incorporate these output values from the circuit. I have used the Rectified linear unit ($ReLU$) activation function to calculate the loss for each output function.

$$ReLU(x) = max\{x, 0\} \tag{2.1}$$

$$\text{Loss } S_{11} = ReLU(\text{ Simulation } S_{11} - \text{ Reference } S_{11}\text{ }) \tag{2.2}$$

$$\text{Loss gain } = ReLU(\text{ Reference gain } - \text{ Simulation gain }) \tag{2.3}$$

$$\text{Loss } NF = ReLU(\text{ Simulation } NF - \text{ Reference } NF\text{ }) \tag{2.4}$$

$$\text{Loss } IIP_3 = ReLU(\text{ Reference } IIP_3 - \text{ Simulation } IIP_3\text{ }) \tag{2.5}$$

$$\text{Loss Idd } = \text{Idd} \tag{2.6}$$

$$\text{Loss} = A_1{\cdot}\text{Loss } S_{11} + A_2{\cdot}\text{Loss gain} + A_3{\cdot}\text{Loss } NF + A_4{\cdot}\text{Loss } IIP_3 + A_5{\cdot}\text{Loss Idd} \tag{2.7}$$

The gain of the circuit (simulation gain) must be above the gain requirement specified (reference gain). Hence, the activation for the gain function is defined as $ReLU($ Reference gain $-$ Simulation gain $)$. The loss function for gain will become positive only when the gain goes below the reference value. The loss function for S_{11} is defined as $ReLU($ Simulation $S_{11} -$ Reference S_{11} $)$. The loss function for S_{11} will become positive only when the S_{11} value goes above the reference value. Similarly, The loss function for the noise figure will become positive only when the noise figure value goes above the reference value. The loss function for IIP_3 will become positive only when the IIP_3 value goes below the reference value. We would like the power consumption in the circuit to ideally be zero. However, in practice, the current drawn from the source cannot go to zero. Hence, the current consumption is directly taken into account when calculating loss.

A_1, A_2, A_3, A_4 and A_5 in equation 2.7 are called as **Loss weights**. I have chosen loss weights for each component of the loss function based on each output value's relative importance and contribution to the loss function. For example, if A_1 is set more than A_2, any deviation in S11 from the reference value will contribute more to the loss function than the same deviation in gain.

2.2 Computing the Gradient of the Loss Function

The circuit variables are incremented or decremented in each iteration based on whether the loss function decreases or increases on the slight increase in each circuit parameter used for optimisation. The direction in which the change must occur is dictated by the gradient of the loss function. This is the fundamental concept applied in the gradient descent algorithm. Since a *ReLU* activation function is utilised, only outputs whose specification is not met in a particular iteration will contribute to the slope updation. Since the gradient of the loss function cannot be computed analytically, there are different numerical approaches. To find the slope of the loss function with respect to a circuit parameter x_j, I am finding the change in the loss function for a small increase in the corresponding circuit parameter, Δx_j. If this change is denoted by $\Delta Loss$, then the slope is given by,

$$\frac{\partial Loss}{\partial x_j} \approx \frac{\Delta Loss}{\Delta x_j} \tag{2.8}$$

The circuit parameters used in the k^{th} iteration of the optimisation are updated using the following formula:

$$x_j(k+1) = x_j(k) - \alpha_j \frac{\partial Loss}{\partial x_j} x_j^2(k) \tag{2.9}$$

The gradient of the loss function is multiplied by the square of the circuit parameter, x_j, to ensure the change is in the same order as the circuit parameter itself. α_j is a **tunable hyperparameter** called **learning rate**, which I have set after running the optimisation for the given netlist more than once. These hyperparameters are crucial in dictating how fast or slow the gradient descent algorithm reaches the local minimum of the loss function. If α_j is very large, the loss function may miss the local minimum and can begin to oscillate or increase. If α_j is too small, arriving at the minimum value will take very long.

Chapter 3

Voltage Mode Passive Mixer Circuit

3.1 Circuit Topology

The **double-balanced, fully differential passive downconversion voltage mode mixer** translates voltage at the RF port (f_{RF}) to diferential output voltage at the baseband (f_{BB}). In the schematic I have chosen for optimisation (given in figure 3.1), the balanced RF inputs (V_{RF}^+ and V_{RF}^-) are each connected to 4 arms of the mixer. Each arm contains a switching transistor connected in series with a passive load (consisting of a resistor, R_B and capacitor C_L). Ideal square-wave quadrature LO waveforms control the transistors with a 25% duty cycle (figure 3.2). The output of this mixer has 2 sets of fully differential voltages at the baseband. These are the in-phase ($V_I^+ - V_I^-$) and quadrature ($V_Q^+ - V_Q^-$) components of the message signal.

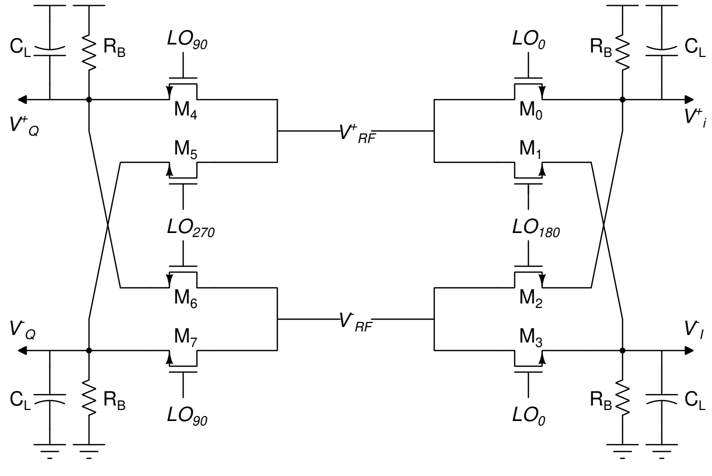


Figure 3.1: Double-balanced voltage mode passive circuit diagram

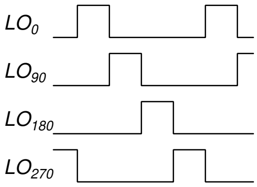


Figure 3.2: Quadrature LO waveforms with 25% duty cycle

3.1.1 Small Signal Analysis

The ideal small signal model of the single-balanced voltage mode mixer is shown in figure 3.3. The circuit is linear and periodically time-varying (LPTV) because the switches are controlled by periodic LO signals. The transistors are approximated to behave as ideal switching devices with a small on-resistance, R_{sw}. When the transistor turns off, the load capacitor holds the voltage at the output node during the switch-off period (discharging happens very slowly for 75% of the LO period). When the transistor switches on, the output voltage roughly tracks the RF signal at the input. The conversion gain of this mixer is equal to the harmonic transfer function from f_{RF} to $f_{RF} - f_{LO}$, denoted by $H_{-1}(j2\pi f_{RF})$, also called as the -1^{th} order harmonic transfer function of the LPTV network. Despite having no amplifying devices in the circuit, this mixer architecture gives a voltage gain slightly greater than unity. R_B helps match the mixer's input impedance. It also helps boost the gain higher than in the case of a purely capacitive load.

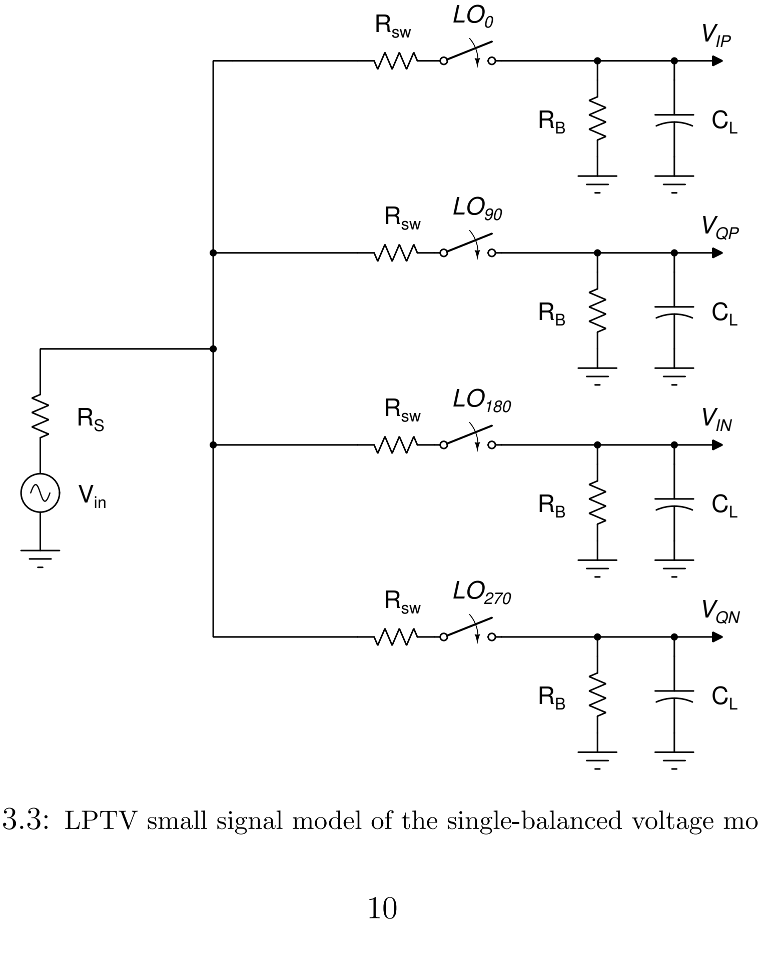


Figure 3.3: LPTV small signal model of the single-balanced voltage mode mixer

There are numerous advantages to choosing the duty cycle to be 25%,

1. 25% duty cycle provides a higher gain than the traditional 50% duty cycle square-wave LO signalling. This is because the magnitude of the first harmonic in the Fourier series of a 25% duty cycle periodic square pulse is $\frac{2\sqrt{2}}{\pi}$, which is 3 dB higher than the magnitude of the fundamental tone in a 50% duty cycle periodic square wave.

2. A 25% duty cycle choice helps match the input impedance to 50Ω. The equation pertaining to input impedance matching is given in the design procedure.

3. The I and Q arms of the mixer are driven by LO_0, LO_{180} and LO_{90}, LO_{270} respectively. Since these waveforms are not simultaneously on, the mixer contributes less to noise and nonlinearity.

The analysis carried out to for the single-balanced mixer holds valid for the fully differential double balanced mixer as well. The small signal analysis for a double-balanced must be done using the differential half circuit to arrive at the same results.

3.2 Design Procedure

The first step in automating schematic design is building a circuit close to satisfying the user's specifications. The netlist corresponding to this circuit design is called the preliminary netlist. The circuit variables input to the gradient descent function as parameters to be tuned are manually calculated in the preliminary netlist. The optimisation algorithm iterates on the preliminary netlist to find the best solution. If the preliminary netlist is closer to the specified output parameters of the circuit, then optimisation is over fast.

For this project, **TSMC 65nm** PDK was used to design the voltage mode passive mixer circuit. The design shown in figure 3.1 was built using Cadence Virtuoso software. The input impedance (Z_{in}) looking in from the differential inputs is matched to 50Ω (differential $Z_{in} = 100\Omega$). The circuit is operated for LO frequencies ranging from 100 MHz to 1 GHz. The bandwidth of the input RF signal is 20 MHz (10 MHz on each side of the LO frequency). The oscillator input is fed to the gate of the switch through a buffer built using inverters (figure 3.4). The design parameters tuned in the optimisation are the resistor's width (W_r), the capacitor's width (W_c) and length (the width and length are equal in all capacitors), the width of the switching nMOSFETs (W_{sw}) and the inverter size ratio (ρ). The nMOSFET size, resistance and capacitance are set to their starting values using the input matching condition. The equations that govern the choice of initial circuit parameters are derived from the small signal, LTI model of the single-balanced mixer given in figure 3.5.

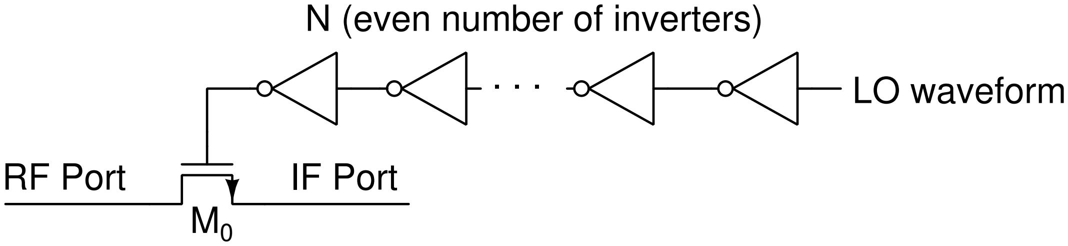


Figure 3.4: Buffer connecting LO input to the switch

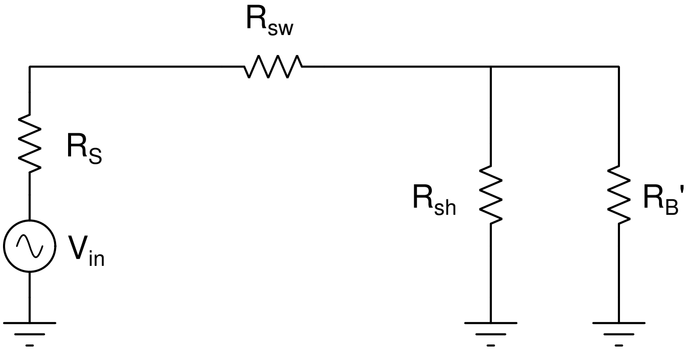


Figure 3.5: LTI small signal model[1], $R_B' = \gamma R_B$

$$R_{sw} \cdot C_L \ll T_{on} \text{ , during the on-time of the switch} \tag{3.1}$$

$$R_B \cdot C_L \gg T_{off} \text{ , during the off-time of the switch} \tag{3.2}$$

$$R_{sh} = (R_{sw} + R_s) \cdot \frac{4\gamma}{1 - 4\gamma} \tag{3.3}$$

$$Z_{in} = R_{sw} + (\gamma R_B) \parallel R_{sh} \tag{3.4}$$

Note, the input port resistance $R_s = 50\Omega$, the on-time of the switch in one LO period is T_{on} and the off-time of the switch in one LO period is T_{off}. The LTI model in figure 3.5 accounts for the LPTV effects of the switches in figure 3.3 with an impedance transform term γ ($= \frac{2}{\pi^2} \approx 0.203$) acting on R_B, and an additional resistance R_{sh}, in shunt with R_B [1]. The value of γ is dependent on the duty cycle and is equal to $\frac{2}{\pi^2}$ when quadrature LO with 25% duty cycle is used.

The smallest inverter size in the buffer is fixed, and the ratio of widths from one inverter to the next is given by ρ. Let C_i denote the input capacitance of the smallest inverter in the buffer and C_{sw} denote the load capacitance of the gate of the switching nMOSFET. The number of inverters in the buffer chain, N, is always an even number and is calculated as,

$$N = \log_{\rho}\left(\frac{2C_{sw}}{C_i}\right) \text{ rounded off to the nearest even integer} \tag{3.5}$$

The equations 3.1 to 3.4 are valid under the assumption that $f_{RF} \approx f_{LO}$. Using this approximation, the contribution to the input impedance by capacitance C_L at baseband is neglected [1]. This assumption holds valid in our case since the bandwidth, $B \ll f_{LO}$ over the range of operation.

With the help of the above equations, the circuit component values that I set in the preliminary netlist are $R_B = 266\Omega$, $C_L = 52.5\ pF$, $W_{sw} = 99\mu m$ and $\rho = 2$.

3.3 Python Optimisation

The gradient descent algorithm for the voltage mode passive mixer circuit minimises the loss function defined in section 2.1. Below is the outline of the routine the Python code follows to carry out the optimisation.

1. The mixer circuit must satisfy the specifications for all LO frequencies from 100 MHz to 1 GHz. The loss function concerning each output is computed at equally spaced LO frequencies within the range. In such a computation, the loss function minimises only when the specifications are met at each LO frequency. The number of frequency points I have chosen is 3. Hence, the outputs are computed at 100 MHz, 550 MHz and 1 GHz to calculate the loss function. The choice of the number of frequency points is up to the designer. For example, the loss function for S11 is,

$$\text{Loss } S_{11} = \sum_{f_{LO}} (\text{Loss } S_{11}@f_{LO}),\ f_{LO} = \{100 \text{ MHz}, 550 \text{ MHz}, 1 \text{ GHz}\} \tag{3.6}$$

   The expressions for the loss concerning noise figure, voltage gain, IIP_3 and power are similar to equation 3.6.

2. I have computed the circuit outputs (S_{11}, gain, noise figure, IIP_3 and power) in every iteration by simulating the netlist. Running the Spectre simulator on Cadence Virtuoso is carried out by calling the command '`spectre <netlist>`' from Python.

3. After each successful circuit simulation, the results are stored in a folder called "`psf`" in the same directory as the netlist. The output is extracted from the `psf` folder using an ocean script executed from Python. The ocean commands

post-process the data stored in the `psf` folder.

4. In every iteration, the gradient of the loss function for each circuit input parameter is calculated by incrementing one circuit parameter at a time, finding the output values and evaluating $\Delta Loss$. Steps 2 and 3 are executed every time the gradient is calculated with respect to the optimising variables.

5. The circuit parameters are updated as
   If Loss $>$ 0

$$W_r \leftarrow W_r - \alpha \frac{\partial Loss}{\partial W_r} W_r^2 \tag{3.7}$$

$$W_c \leftarrow W_c - \alpha \frac{\partial Loss}{\partial W_c} W_c^2 \tag{3.8}$$

$$W_{sw} \leftarrow W_{sw} - \alpha \frac{\partial Loss}{\partial W_{sw}} W_{sw}^2 \tag{3.9}$$

$$\rho \leftarrow \rho - \alpha \frac{\partial Loss}{\partial \rho} \rho^2 \tag{3.10}$$

   In the above equations, α is the learning rate in the optimisation. W_r, W_c and W_{sw} are the widths of the resistance R_B, capacitance C_L and nMOSFET switch respectively. I have chosen the learning rate to be 0.01. This ensures that the number of inverters (N) does not update to a negative integer in any of the iterations. The former is possible if ρ changes by a large step within a single iteration.

6. When the contribution to loss in equation 2.7 is only due to the current loss from equation 2.6, the optimisation algorithm increases the learning rate for faster convergence. Since the current consumption is only by the buffers in the circuit, the Python code also checks if a decrease or increase in the number of inverters (N) can reduce the power before increasing the learning rate.

3.4 Results

	Pre-optimisation	Specification	Post-optimisation
S_{11}	-23.98 dB	< -15 dB	-16.05 dB
Gain	2.61 dB	> 4 dB	4.27 dB
Integrated SSB Noise figure	6.86 dB	< 7 dB	6.01 dB
IIP_3	24.78 dBm	> 15 dBm	26.42 dBm
Power	3.692 mW	Minimise	3.529 mW

Table 3.1: Specifications at $f_{LO} = 550$ MHz before and after optimisation

Table 3.1 shows the specifications at $f_{LO} = 550$ MHz before and after running the Gradient descent algorithm using Python.

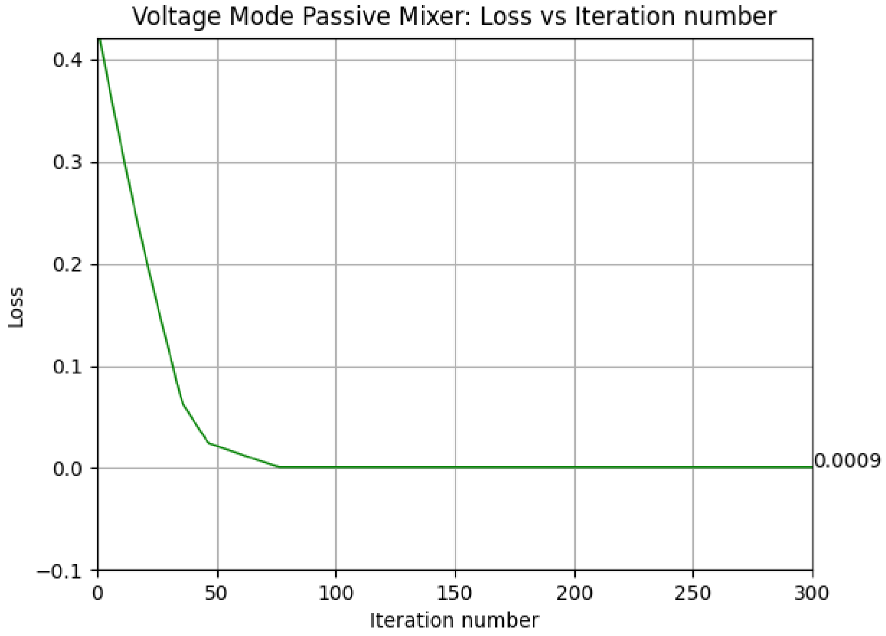


Figure 3.6: Loss as a function of number of iterations

Figure 3.6 shows the loss function's variation over 300 iterations. The loss decreases with increasing number of iterations. The circuit parameters after optimisation are $R_B = 263.63\Omega$, $C_L = 110\ pF$, $W_{sw} = 96.13\mu m$ and $\rho = 3.52$.

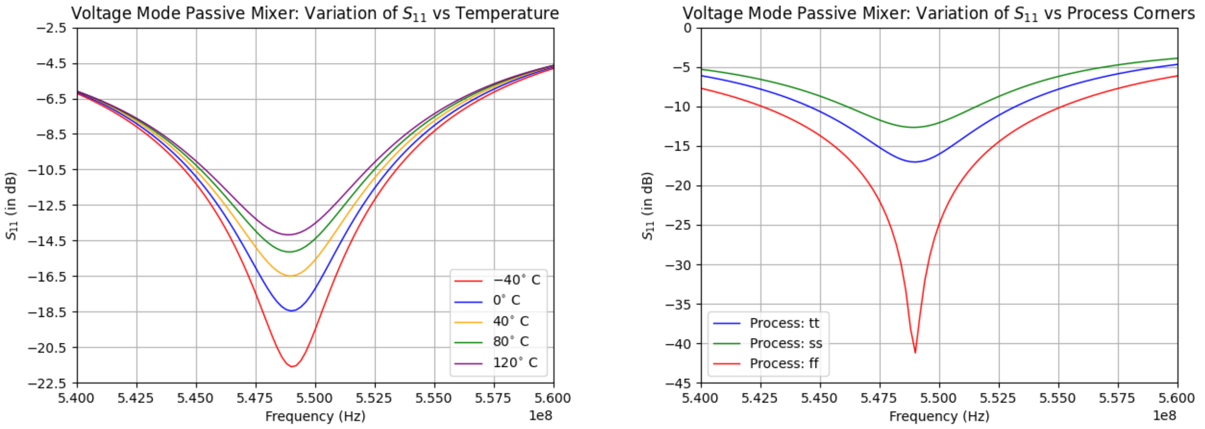
Loss	Starting value	Post-optimisation
S_{11}	0	0
Gain	0.421	0
SSB NF	0	0
IIP_3	0	0
Power	9.23×10^{-4}	8.82×10^{-4}
Total loss	0.422	8.82×10^{-4}

Table 3.2: Loss values before and after optimisation

Table 3.2 shows the value of the loss function before and after optimisation. The loss function incorporates specifications at 100 MHz, 550 MHz and 1 GHz as per equation 3.6.

3.4.1 Process and Temperature Variations

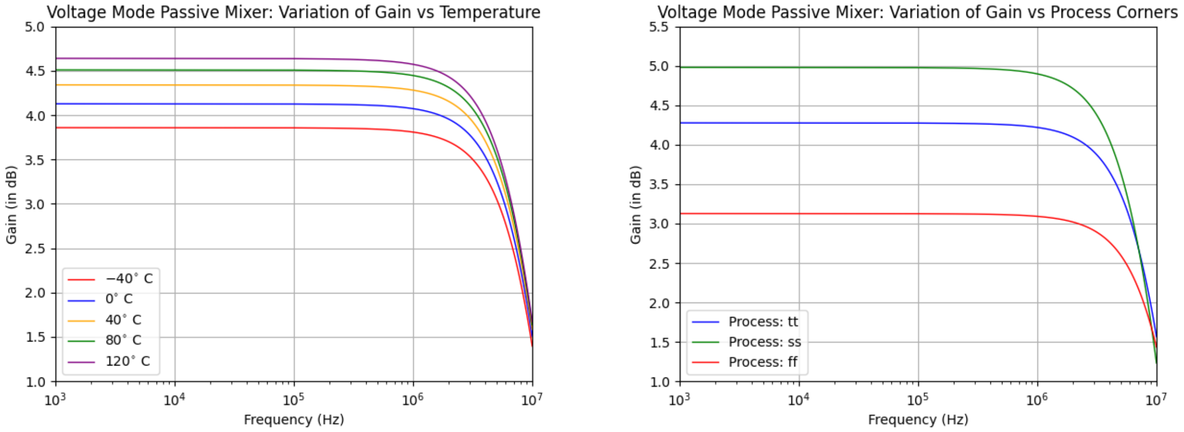
After optimization, the circuit is tested for temperature and process variations. The following analysis is carried out at the following temperatures - [−40°C, 0°C, 40°C, 80°C, 120°C] and the following process corners - [ss (slow slow), tt (typical typical), ff (fast fast)].



(a) Temperature

(b) Process variations at 27°C

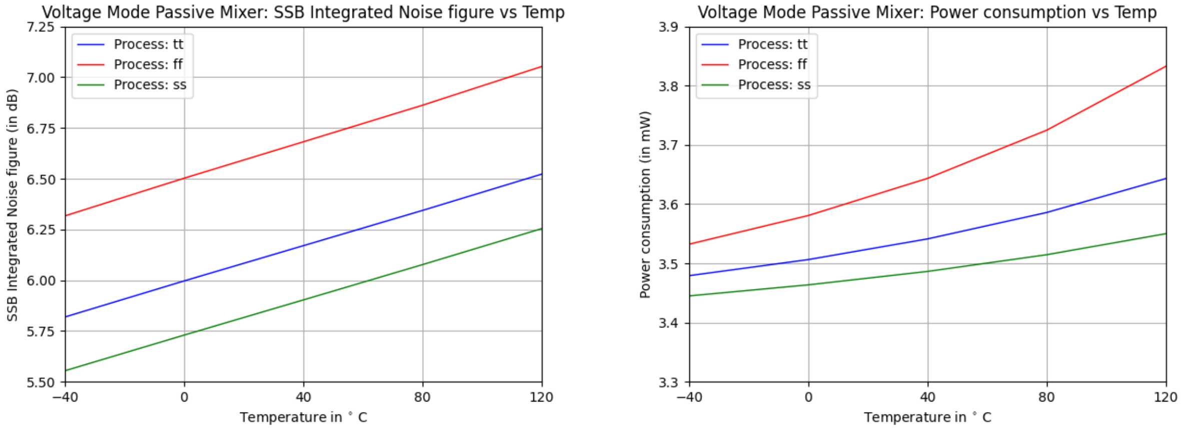
Figure 3.7: S_{11} - process and temperature variations



(a) Temperature

(b) Process variations at 27°C

Figure 3.8: Gain - process and temperature variations



(a) Integrated SSB Noise figure

(b) Power consumption

Figure 3.9: Noise figure and power consumption - variations with process and temperature

From figure 3.7a and 3.7b, we observe that S_{11} is the largest when the temperature is 120°C and when the process is ss (slow slow) respectively. At this temperature and process, the input impedance matching is the poorest. From figure 3.8a and 3.8b, we observe that voltage gain reaches the lowest value when the temperature is -40°C and when the process is ff (fast fast) respectively. From figure 3.9a, we observe that SSB integrated noise figure is highest (= 7.05 dB) in the ff (fast fast) process corner at 120°C. From figure 3.9b, we observe that power consumption is highest (= 3.83 mW) in the ff (fast fast) process corner at 120°C.

Chapter 4

Current Mode Passive Mixer Circuit

4.1 Circuit Topology

Traditional RF receivers are almost always built with Low-Noise Amplifiers (LNAs) at the front end of the receiver chain, followed by a mixer that downconverts RF (f_{RF}) to baseband (f_{BB}). The LNA provides a significant gain to the received RF voltage (usually in the order of μV) while adding a minimum noise figure to the rest of the receiver. In cases where the LNA has a relatively large output impedance (figure 4.1), approximating a current source, the input to the mixer is no longer an RF voltage. Here, the input to this type of mixer is an RF current source.

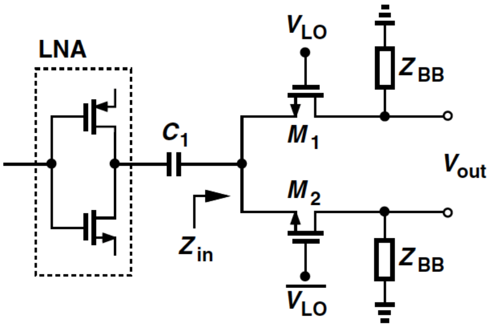


Figure 4.1: Conceptual arrangement where the LNA has relatively high output impedance ref: RF Microelectronics, *Razavi*

The **double-balanced, fully differential current mode downconversion mixer** has a balanced input RF current and outputs differential baseband voltage. Quadrature LO waveforms with 25% duty cycle are used, and 2 sets of differential output voltages are the I and Q components of the message signal, respectively.

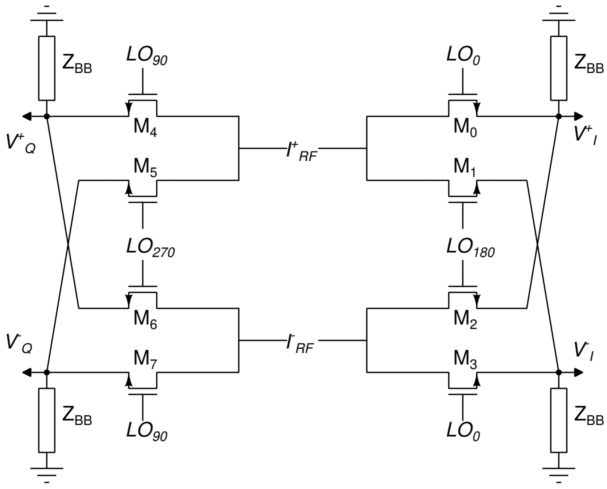


Figure 4.2: Double-balanced current mode passive mixer, Z_{BB} represents a trans-impedance amplifier

In the schematic I have chosen for optimisation (figure 4.2), each differential input is connected to 4 arms of the mixer, each presenting a switching transistor connected in series with a **trans-impedance amplifier (TIA)**. The trans-impedance amplifier (figure 4.3) translates the switching current from the RF side to a baseband output voltage. The trans-impedance, $Z_{BB}(j2\pi f)$ (or current to voltage gain) of the amplifier is given by (the OPAMP is assumed to be ideal)

$$Z_{BB}(j2\pi f) = \frac{V_{out}}{I_{in}} = \frac{R_L}{1 + j2\pi f C_L R_L} \tag{4.1}$$

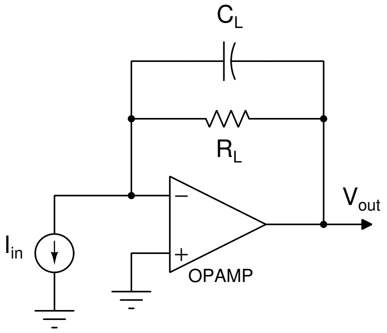


Figure 4.3: trans-impedance amplifier

Though a very similar design, the current-mode mixer differs from the voltage-mode counterpart in many ways.

1. The input impedance of the current-mode passive mixer must be **near zero** to ensure almost all current flows into the mixer stage rather than the output impedance of the LNA.

2. The input current at RF is converted to a voltage at the baseband by the

nature of the trans-impedance amplifier at the load. The conversion gain from the input RF current to the output voltage is measured as trans-impedance (Ω).

In each arm of the mixer, the input current flows through the baseband amplifier, denoted by Z_{BB}, for 25% of the LO period. This can be thought of as the input current signal multiplied by an ideal periodic square pulse of amplitude 1 with the same period and duty cycle as the LO signal. The baseband component of this current is subjected to the amplifier's frequency response and upconverted to the input RF port by the nature of the mixing circuit (figure 4.4). This implies that the input impedance looking into the mixer from the differential RF port resembles a frequency-translated version of $Z_{BB}(j2\pi f)$. Therefore, the trans-impedance amplifier affects the input impedance and the gain of the current mode mixer.

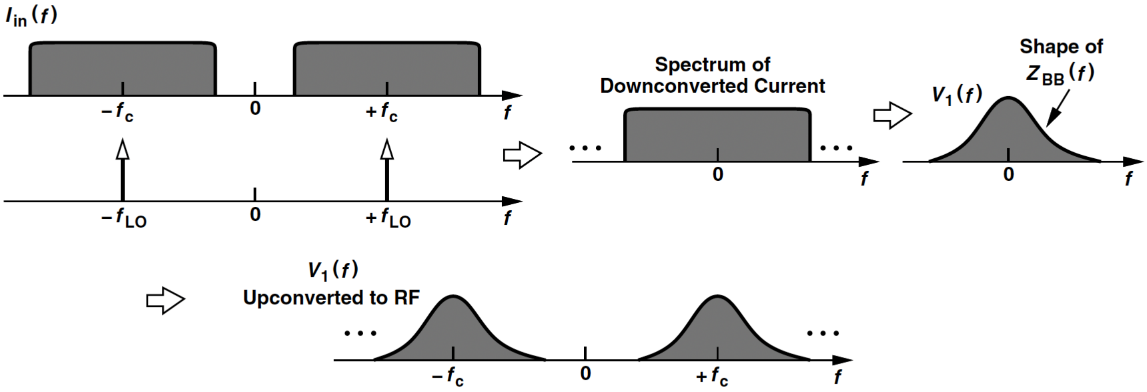


Figure 4.4: frequency translation of $Z_{BB}(j2\pi f)$ ref: RF Microelectronics, *Razavi*

4.2 Design Procedure

In this project, **TSMC 65nm** PDK is used in building the current mode passive mixer. The circuit shown in figure 4.2 was built using Cadence Virtuoso. The circuit is operated for LO frequencies ranging from 100 MHz to 1 GHz. The bandwidth of the input RF signal is 20 MHz (10 MHz on each side of the LO frequency). The oscillator input is fed to the gate of the switch through a buffer built using inverters (similar to the voltage mode mixer, figure 3.4). The input impedance of the mixer looking in from the differential RF inputs is set close to 0 ohms ($S11 = 0$dB) to ensure that less than 5% of the current flows through the output impedance of the LNA. The loss function is defined using equation 2.7, section 2.1. While simulating the mixer circuit to find input impedance, gain, noise figure, IIP_3 and power consumption, practical values are chosen for the gain of the LNA stage and the gain of OPAMP used in the trans-impedance amplifier.

In this project, the trans-conductance of the LNA $g_m = 40mS$, the output impedance of the LNA, $R_N = 250\Omega$ and the open loop gain of the OPAMP used in the trans-impedance amplifier $G = 100$. The optimizing variables in the circuit are the width of the resistor used in the TIA (W_r), the width and length of the capacitor used in the TIA (W_c), the width of the switching nMOSFETs (W_{sw}) and the inverter size ratio (ρ). The starting values I have chosen for the circuit are $R_L = 3K\Omega$, $C_L = 10\ pF$, $W_{sw} = 99\mu m$ and $\rho = 2$.

The Python optimisation routine follows the steps in section 3.3. Once again, the loss function is computed at evenly spaced frequency points from 100 MHz to 1 GHz.

4.3 Results

	Pre-optimisation	Specification	Post-optimisation
S_{11}	-5.66 dB	$\|S_{11}\| < 5$ dB	-4.953 dB
Gain	61.847 dB	> 60 dB	60.532 dB
Integrated SSB Noise figure	7.28 dB	< 8 dB	7.3 dB
IIP_3	25.63 dBm	> 15 dBm	27.46 dBm
Power	12.431 mW	Minimise	10.09 mW

Table 4.1: Specifications at $f_{LO} = 550$ MHz before and after optimisation

Table 4.1 shows the specifications at $f_{LO} = 550$ MHz before and after running the Gradient descent algorithm using Python.

Figure 4.5 shows the loss function's variation over 300 iterations. The loss decreases with increasing number of iterations. The lowest loss was recorded after **78** iterations. The loss function continues to oscillate after that to optimise the power consumption. The circuit parameters after optimisation are $R_L = 2.4445K\Omega$, $C_L = 8.58\ pF$, $W_{sw} = 107.2\mu m$ and $\rho = 1.93$.

Loss	Starting value	Post-optimisation
S_{11}	0.098	0
Gain	0	0
SSB NF	0	0
IIP_3	0	0
Power	3.1×10^{-3}	2.523×10^{-3}
Total loss	0.101	2.523×10^{-3}

Table 4.2: Loss values before and after optimisation

Table 4.2 shows the value of the loss function before and after optimisation.

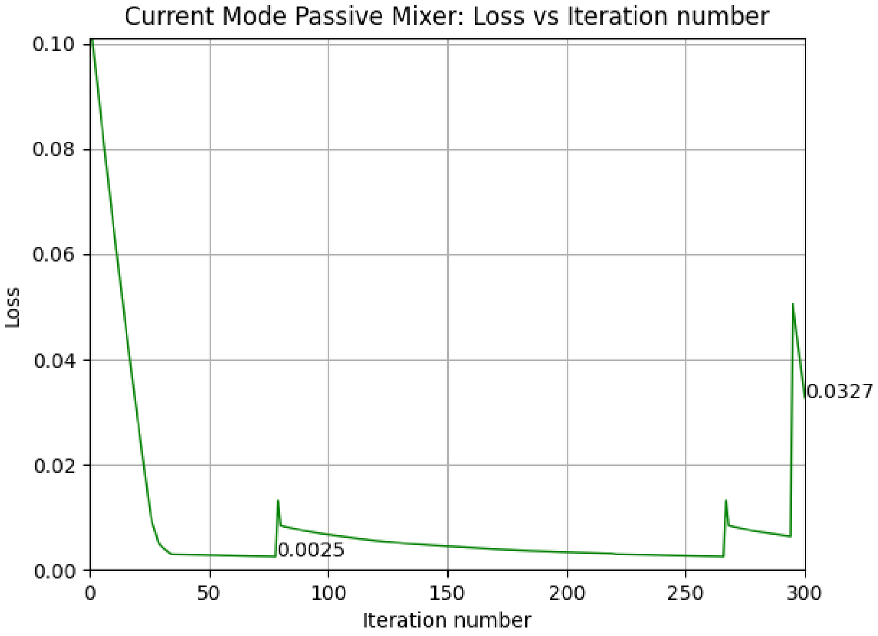
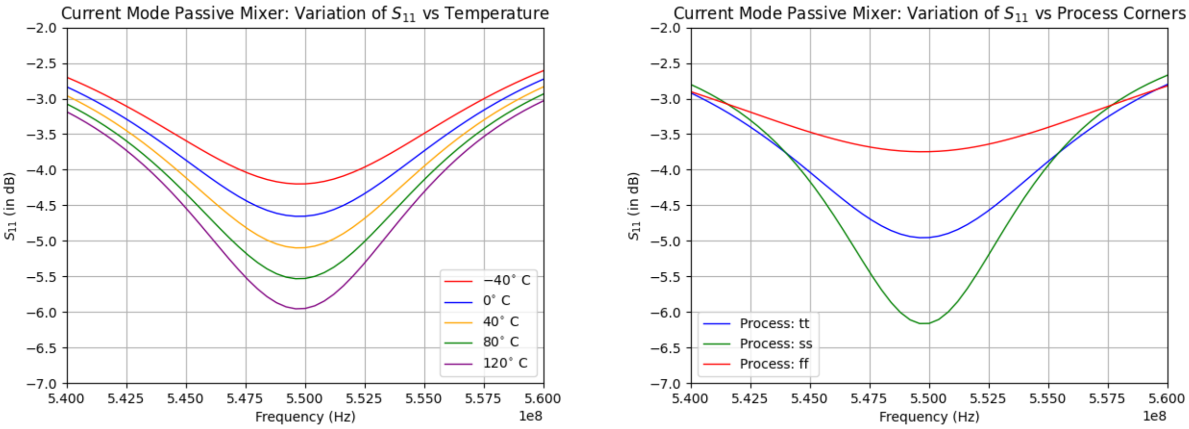


Figure 4.5: Loss as a function of number of iterations

4.3.1 Process and Temperature Variations

After optimization, the circuit is tested for temperature and process variations. The following analysis is carried out at the following temperatures - [−40°C, 0°C, 40°C, 80°C, 120°C] and the following process corners - [ss (slow slow), tt (typical typical), ff (fast fast)].

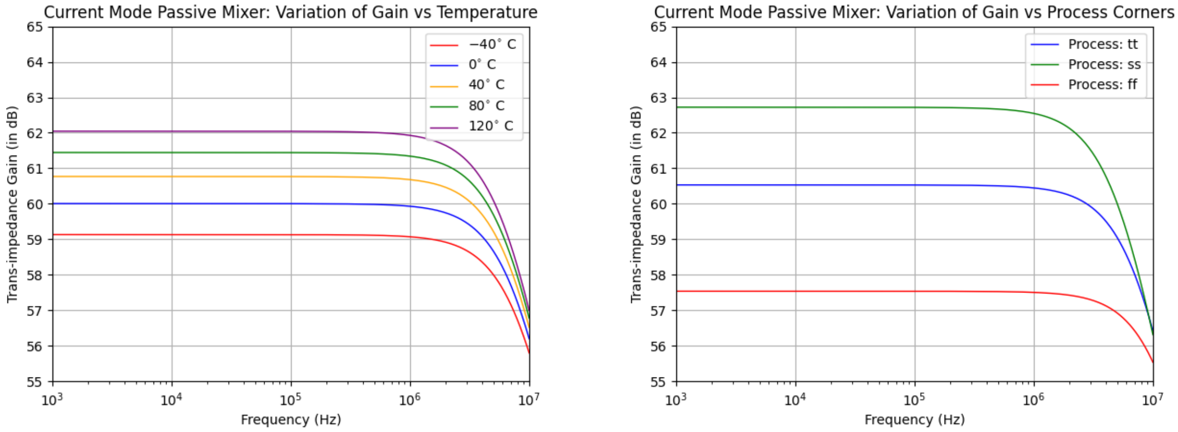


(a) Temperature

(b) Process variations at 27°C

Figure 4.6: S_{11} - process and temperature variations

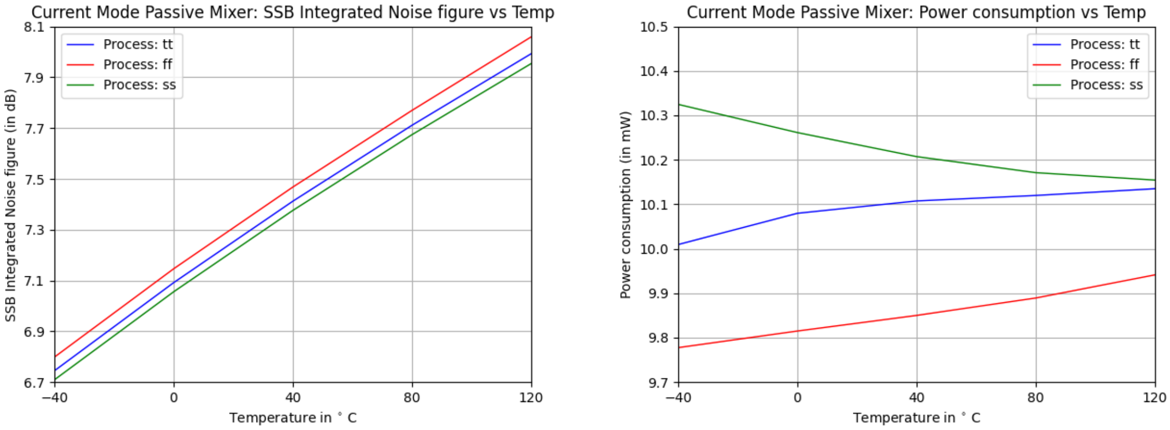
From figure 4.6a and 4.6b, we observe that S_{11} is the least (more negative) when the temperature is 120°C and when the process is ss (slow slow) respectively. At this temperature and process, the input impedance matching is the poorest.



(a) Temperature

(b) Process variations at 27°C

Figure 4.7: Gain - process and temperature variations



(a) Integrated SSB Noise figure

(b) Power consumption

Figure 4.8: Noise figure and power consumption - variations with process and temperature

From figure 4.7a and 4.7b, we observe that voltage gain reaches the lowest value when the temperature is −40°C and when the process is ff (fast fast) respectively. From figure 4.8a, we observe that SSB integrated noise figure is highest (= 8.06 dB) in the ff (fast fast) process corner at 120°C. From figure 4.8b, we observe that power consumption is highest (= 10.325 mW) in the ss (slow slow) process corner at −40°C.

Chapter 5

Conclusion

In this project, the schematic design automation of voltage mode and current mode passive downconversion mixers was carried out. The gradient descent algorithm was used to achieve the best set of circuit parameters that satisfy the specification. The circuit chosen for automatic synthesis was the double-balanced, fully differential mixer topology. The LO waveforms are input to the mixer through buffers built using inverters. After Python optimisation, the code performs analysis over temperature and process variations.

5.1 Further Improvements

- The Python code only optimises the circuit whose netlist has already been generated from design. The code functions only on two types of passive double-balanced mixers. The code does not generate a netlist from scratch. **Automatic RFIC Synthesis** can be implemented by including more types of mixer topologies and creating a database of circuit designs from which the user can choose the best performing circuit. The Python optimisation algorithm can be generalised to automate any given Mixer circuit for a set of design specifications.

- In this project, only the schematic design has been automated. The Python code framework can be extended to automate the layout design of the circuit as well.

- The local minima achieved in gradient descent depends on the choice of initial circuit parameters in the netlist. Hence the **role of a circuit designer is very important** in converging to the optimum solution faster.

Bibliography

[1] C. Andrews and A. C. Molnar. Implications of Passive Mixer Transparency for Impedance Matching and Noise Figure in Passive Mixer-First Receivers. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(12):3092–3103, December 2010.

[2] C. Andrews and A. C. Molnar. A Passive Mixer-First Receiver with Baseband-Controlled RF Impedance Matching, < 6 dB NF, and > 27 dBm Wideband IIP_3. *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, 53:46–47, February 2010.

[3] Behzad Razavi. *RF Microelectronics*. Prentice Hall, second edition edition, 2011.