LUNDS UNIVERSITET

Lunds Tekniska Högskola

Department of Electrical and Information Technology

Simulations of III-V NWFET Double-Balanced Gilbert Cells with an Improved Noise Model

---

Author: Niklas Lindblad

Examiner: Johannes Svensson

Supervisors:
Karl-Magnus Persson and Lars-Erik Wernersson

May 30, 2013

Abstract

III-V nanowire transistors might provide a mean for extending Moore's law, by overcoming the scaling limitations ultimately facing planar silicon CMOS. These high frequency capable transistors with cut-off frequencies in the terahertz regime are suitable for radio communication. In this project an active double-balanced gilbert cell mixer consisting of nanowire field-effect transistors (NWFETs) was simulated in Cadence Virtuoso using a compact transistor model. The transistor model was extended to take flicker and thermal noise into account, in order to more accurately compare the mixers against state-of-the-art silicon CMOS implementations.

The final mixer for 60 GHz showed much greater linearity (0.4 dBm 1 dB compression and 8.5 dBm $IIP3$) than previously reported silicon CMOS counterparts. It exhibited a conversion gain of 3.47 dB, a NF_{DSB} of 14.6 dB and a DC power consumption of 8.7 mW. Based on these findings the design requirements for suitable low noise amplifier was discussed.

Acknowledgements

I would like to thank my supervisors Karl-Magnus Persson and Lars-Erik Wernersson, for their enthusiasm and guidance throughout this thesis. It has been wonderful to attend your weekly meetings and get a glimpse of your research in a unique way.

I also want to thank my parents Bo and Eva Lindblad for their support and encouragement. Showing their eight year old son π when he exclaimed the stupidity of measuring the circumference of a circle using a string of yarn, as the math book suggested, is just one example.

My close friends Rasmus, Mikael, Nicklas, Alexander, Sebastian and Joseph for listening to what I can only imagine seemed like incoherent rambles about nanowires, noise models and trade-offs in analog design.

Last but not least I would like to thank all the amazing people I have met over the years in the vivid student life of Lund. Thank you Radio AF, The Academic Society and Lundakarnevalen for all the distractions, challenges and friendships.

> "Why cannot we write the entire 24 volumes of the Encyclopaedia Brittanica on the head of a pin?"
>
> Richard P. Feynman, December 29th 1959

Contents

Page

List of Tables **iv**

List of Figures **v**

Aims and Outline of this Thesis **vii**

1 Introduction **1**
1.1 III-V Nanowire Semiconductors . . . 1
1.2 Millimetre Wave Communication . . . 3
1.3 Frequency Limitations of Silicon CMOS . . . 4
1.3.1 Silicon On Insulator . . . 4
1.3.2 SiGe BiCMOS . . . 4

2 Theory **5**
2.1 Mixer Fundamentals . . . 5
2.2 Receiver Topologies . . . 6
2.2.1 Superheterodyne Receivers . . . 6
2.2.2 Homodyne Receivers . . . 6
2.3 Figures of Merit . . . 7
2.3.1 Power Consumption . . . 7
2.3.2 RF to IF Conversion Gain . . . 7
2.3.3 Usable Bandwidth . . . 7
2.3.4 LO to RF and LO to IF Isolation . . . 7
2.3.5 1 dB Compression Point . . . 8
2.3.6 Third-Order Intercept Point (IIP3) . . . 8
2.3.7 Noise . . . 9
2.4 Combined Figures of Merit . . . 10
2.5 Implementation Using Field Effect Transistors . . . 11
2.5.1 Active and Passive Mixers . . . 11
2.6 Different Active Mixer Topologies . . . 12
2.6.1 Single Device (Square-Law Mixer) . . . 12
2.6.2 Single-Balanced . . . 13
2.6.3 Double-Balanced (Gilbert Cell) . . . 15
2.7 Performance Enhancements . . . 18
2.7.1 Source Degeneration . . . 18
2.7.2 G_m-boosting . . . 20
2.7.3 Current Bleeding . . . 21
2.8 SpectreRF Simulations in Cadence Virtuoso . . . 22
2.8.1 Periodic Steady-State Analysis (PSS) . . . 22
2.8.2 Periodic AC Analysis (PAC) . . . 22
2.8.3 Quasi-Periodic Steady-State Analysis (QPSS) . . . 22
2.8.4 Quasi-Periodic AC Analysis (QPAC) . . . 22
2.8.5 Periodic Transfer Function Analysis (PXF) . . . 22
2.8.6 Periodic Noise Analysis (Pnoise) . . . 22
2.9 MOS Transistor Noise . . . 23

2.9.1 Shot Noise . . . 23
2.9.2 Thermal Noise . . . 23
2.9.3 Flicker Noise . . . 23
2.9.4 Noise Spectral Density . . . 23
2.9.5 Modelling Correlated Noise . . . 24

3 Method **25**
3.1 Automated RF Mixer Test Bench in Cadence Virtuoso . . . 25
3.1.1 DC Power Consumption versus LO Signal Power . . . 26
3.1.2 Voltage Conversion Gain versus LO Signal Power . . . 26
3.1.3 Voltage Conversion Gain versus RF Frequency . . . 26
3.1.4 Port Isolation . . . 26
3.1.5 1 dB Compression and Third-Order Intercept . . . 26
3.1.6 Noise Figure versus LO Signal Power . . . 26
3.2 Implementation of Noise Model for the NWFET . . . 27
3.2.1 Noise Contributions . . . 30
3.2.2 Scaling Mixer to Reduce Noise . . . 30
3.3 Revised Transistor Layout for Larger Number of Wires . . . 31
3.4 Design and Implementation of III-V NWFET-based Gilbert Cell . . . 34
3.4.1 Working Gilbert Cell (Mixer A) . . . 35
3.4.2 Source Degeneration (Mixer B) . . . 36
3.4.3 Current Bleeding (Mixer C) . . . 37
3.4.4 Source Degeneration and Current Bleeding (Mixer D & E) . . . 37
3.5 LNA Design Requirements . . . 38

4 Results **39**
4.1 Noise Model Implementation . . . 39
4.1.1 Noise Contributions . . . 39
4.1.2 Scaling Mixer to Reduce Noise . . . 40
4.2 Revised Transistor Layout for Larger Number of Wires . . . 42
4.3 Mixer Design . . . 47
4.3.1 Working Gilbert Cell (Mixer A) . . . 47
4.3.2 Source Degeneration (Mixer B) . . . 48
4.3.3 Current Bleeding (Mixer C) . . . 49
4.3.4 Source Degeneration and Current Bleeding (Mixer D & E) . . . 49
4.4 LNA Design Requirements . . . 50

5 Discussion **52**
5.1 Automated RF Mixer Test Bench in Cadence Virtuoso . . . 52
5.1.1 Future Improvements . . . 52
5.2 Implementation of Noise Model for the NWFET . . . 53
5.2.1 Verification by Calculations . . . 53
5.2.2 Accuracy . . . 54
5.3 Revised Transistor Layout for Larger Number of Wires . . . 54
5.3.1 Maximum Achievable Voltage Conversion Gain . . . 55
5.4 Design and Implementation of III-V NWFET-based Gilbert Cell . . . 56
5.4.1 Working Gilbert Cell (Mixer A) . . . 56
5.4.2 Source Degeneration (Mixer B) . . . 56
5.4.3 Current Bleeding (Mixer C) . . . 57

5.4.4 Source Degeneration and Current Bleeding (Mixer D & E) 57
5.5 LNA Design Requirements 57

6 Conclusions **58**

References **59**

Code **65**

Schematics **84**

List of Abbreviations **88**

List of Tables

1 Frequency capabilities for transistors in different technologies. 4
2 Comparison between active and passive mixers [52, p. 73-90]. 11
3 Differently sized NWFETs with constant $n_{NW,y} = 8$. 31
4 Small-signal resistances and their layout dependence. 32
5 Small-signal capacitances and layout dependencies. 33
6 Summary of the different mixers to design and benchmark. 34
7 $P_{LO} = 7\ dBm$ and $IF = 100\ MHz$. 39
8 Comparing the figures of merits of mixer A with previous work. 56
9 Previously reported 60 GHz silicon CMOS receiver front-ends. 57

List of Figures

1 Schematic of Nanowire Field-Effect Transistor (NWFET). (a) Cross-section and (b) Cut along cylindrical axis. . . . 1
2 Small signal model of the NWFET. . . . 2
3 Small signal model of the NWFET in Cadence Virtuoso. . . . 2
4 Unlicensed frequencies between 57 and 68 GHz (overlap in grey). . . . 3
5 Symbolic representation of an ideal mixer acting as a multiplier. . . . 5
6 Block diagram of typical superheterodyne receiver front-end. . . . 6
7 Block diagram of typical homodyne receiver front-end. . . . 6
8 LO to RF and LO to IF feedthrough. . . . 7
9 Graphical definition of 1 dB compression point. . . . 8
10 Graphical definition of third-order intercept point. . . . 8
11 Dominating blocks in the typical receiver. . . . 9
12 Simplified circuit diagram of a single-balanced mixer. . . . 13
13 The different stages of the single-balanced mixer. . . . 14
14 Simplified circuit diagram of the Gilbert cell. . . . 15
15 The different stages of the gilbert cell. . . . 16
16 Scaling of entire mixer to reduce noise. . . . 17
17 Common-source amplifier with resistive source degeneration. . . . 18
18 Common-source amplifier with inductive source degeneration. . . . 19
19 General G_m-boosting at the RF stage. . . . 20
20 Detailed G_m-boosting at the RF stage, as outlined in [59]. . . . 20
21 Double-balanced Gilbert cell with current bleeding. . . . 21
22 Typical equivalent noise voltage spectral density for a MOSFET [64, p. 763]. . . . 23
23 Overview of test modules and workflow. . . . 25
24 Source of noise in the original small signal model of the NWFET. . . . 27
25 Measured noise data for the NWFET, courtesy of Karl-Magnus Persson. . . . 28
26 Linear regression of measured noise data for the NWFET. . . . 29
27 Source of noise in the new model. . . . 30
28 Distinction between contributions from the new noise source and the previous resistive elements. . . . 30
29 Symmetrical transistor layout with $n_{NW,x}$ close to $n_{NW,y}$. . . . 31
30 Asymmetrical transistor layout with $n_{NW,x} > n_{NW,y}$. . . . 32
31 Proposed design flow. . . . 34
32 Noise output voltage spectrum. . . . 39
33 Device scaling effects on noise and DC power. . . . 40
34 Device scaling effects on voltage conversion gain. . . . 41
35 I_{DS} normalised to total wire circumference plotted against V_{DS}. . . . 42
36 Small-signal resistor values for an asymmetrical scaling case. . . . 43
37 Small-signal resistor values for a symmetrical scaling case. . . . 44
38 I_{DS} normalised to total wire circumference plotted against V_{DS} (improved transistor layout). . . . 45
39 g_m plotted against V_{GS} for $V_{DS} = 0.3\,V$ for both symmetrical and asymmetrical transistor. . . . 46
40 L_{deg} plotted against $IIP3$ and voltage conversion gain. . . . 48
41 LNA design requirements for mixer A. . . . 50
42 LNA design requirements for mixer A with sub-milliwatt power consumption. . . . 51

43 Abstraction layers in the overall design. 52
44 Limiting capacitance exemplified with a single-balanced mixer. 55
45 Schematic for the test bench setup. 84
46 Schematic for mixer A. 85
47 Schematic for mixer B. 86
48 Schematic for mixer C. 87

Aims and Outline of this Thesis

The aim of this project is to do a comparison between radio frequency (RF) mixer circuits implemented using III-V Nanowire Field-Effect Transistors (NWFETs) and traditional silicon CMOS transistors (Si CMOS). Traditionally, benchmarking of new technologies tend to focus on simple circuits (e.g. an inverter for digital IC) since more complex circuits might not yet be possible to realise until the technology becomes more mature. The limits of circuit manufacturing can be circumvented by instead carrying out the benchmarking using simulation software, such as Cadence Virtuoso.

The validity of data acquired from such simulations depend on the quality of the underlying device models. In this thesis the NWFET simulated is based on previous work by the Nano Electronics Group at the Department of Electrical and Information Technology at Lund University, Sweden. The transistor model has been implemented using a virtual-source model and also includes a rigorous model of the capacitances arising from the device structure. Using the well-established SpectreRF component of Cadence Virtuoso, the accuracy of these simulations should be sufficient for comparison with both manufactured and simulated equivalent silicon CMOS mixers found in the literature.

The first part of the project is an investigation of which frequency ranges to consider for the mixer implementation and finding relevant previous work consisting of silicon CMOS based mixers operating at these frequencies. In order to enable further comparisons, the figures of merits of a general mixer circuit are defined, as well as common circuit topologies used. Since the aim of this thesis is to implement a III-V NWFET based mixer circuit of equivalent complexity as state-of-the-art silicon CMOS based mixer circuits, the many ingenious design features presented in the literature needs to be investigated and quantified in able to incorporate them into the final III-V NWFET mixer circuit.

The second part of this project is to develop an automated test bench for mixer circuits in Cadence Virtuoso, in order to simplify future design efforts and assure that the simulations carried out are well documented and highly repeatable. The automation is done using the Open Command Environment for Analysis (OCEAN) in Cadence Virtuoso, based on the SKILL programming language (a dialect of LISP). Together with data analysis in MatLab, this provides a good foundation for future mixer design efforts.

Without a noise model for the transistor an important figure of merit, the noise figure, cannot be simulated. In order to simulate it, the third part of this project focuses on implementing a noise model based on both measured data and well-established analytical expressions.

Finally, a III-V NWFET based active down-converting mixer is to be designed and characterised. Based on the figures of merit for the mixer, the design constraints for a suitable low noise amplifier (LNA) for further implementation of a receiver or transmitter can be discussed.

1 Introduction

1.1 III-V Nanowire Semiconductors

Almost four decades ago, G.E. Moore revised the prediction of his famous law to read: "The number of transistors per chip will double every 18 months." [1]. Although the device scaling necessary to follow this exponential projection has been made possible by advancements and refinements in semiconductor fabrication, traditional planar silicon CMOS devices face a variety of difficulties in scaling to 10 nm gate length and below. Tunnelling limits the thickness of gate oxides and gate length scaling of bulk metal-oxide semiconductor field-effect transistors (MOSFETs) introduces unacceptable short-channel effects [2, 3].

Nanowire transistors have gained much attention in research as a possible replacement to the traditional planar CMOS structures, since advanced structures such as cylindrical nanowire MOSFETs with gate all around provide better electrostatic integrity, which gives greater control over the short-channel effects [3, 4]. By also changing material from silicon to III-V compounds, such as indium arsenide (InAs), the performance increases even further due to the superior electron transport properties [5]. The structure of these devices is described in *figure 1*, the small signal model in *figure 2* and the Cadence Virtuoso small signal equivalent in *figure 3*.

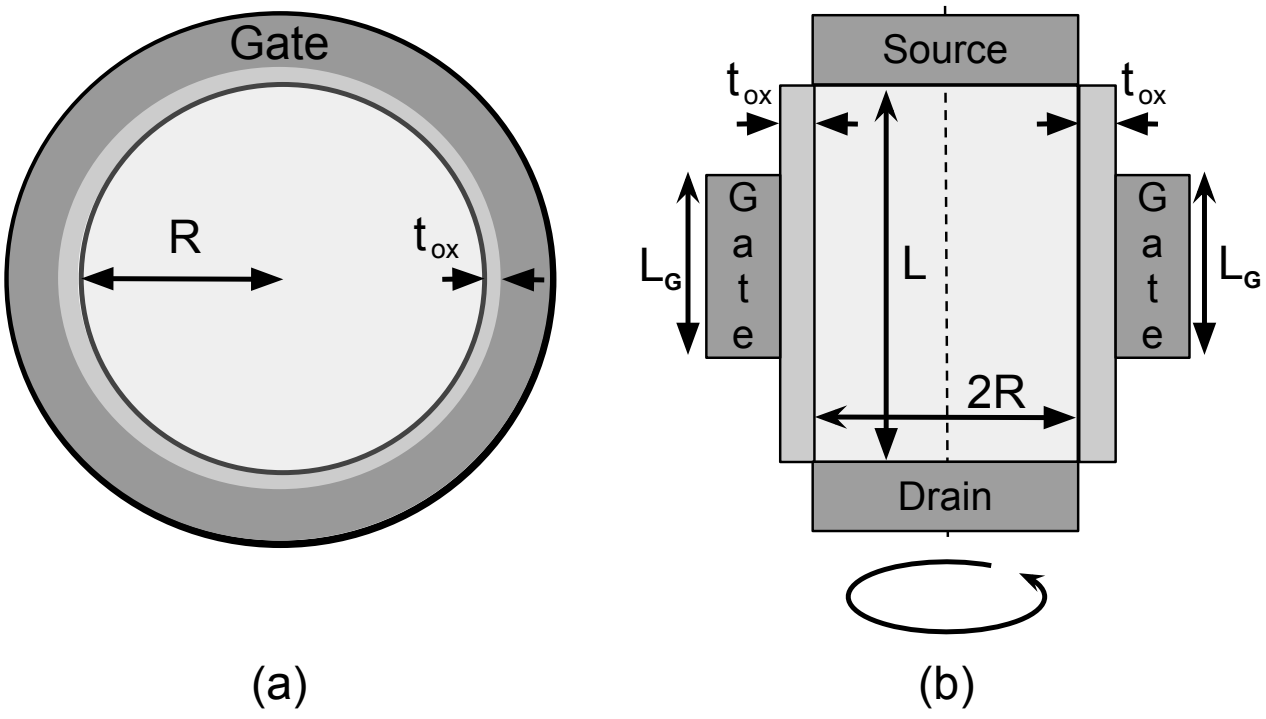

Figure 1: Schematic of Nanowire Field-Effect Transistor (NWFET). (a) Cross-section and (b) Cut along cylindrical axis.

The NWFETs considered in this project (60 nm gate length and 45 nm diameter) are InAs vertical nanowire arrays optimised for radio frequency applications [6, 7] as part of previous work by the Nano Electronics Group (Department of Electrical and Information Technology at Lund University, Sweden.). These devices can be manufactured on silicon substrates and the nanowires are defined by electron beam lithography patterning of gold particles [5].

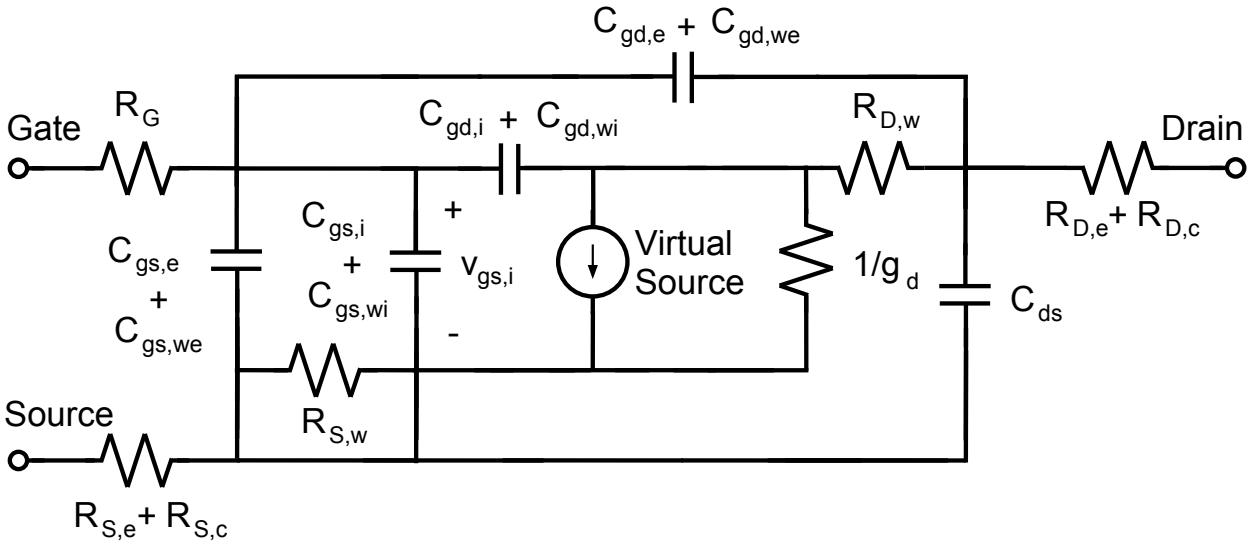

Figure 2: Small signal model of the NWFET.

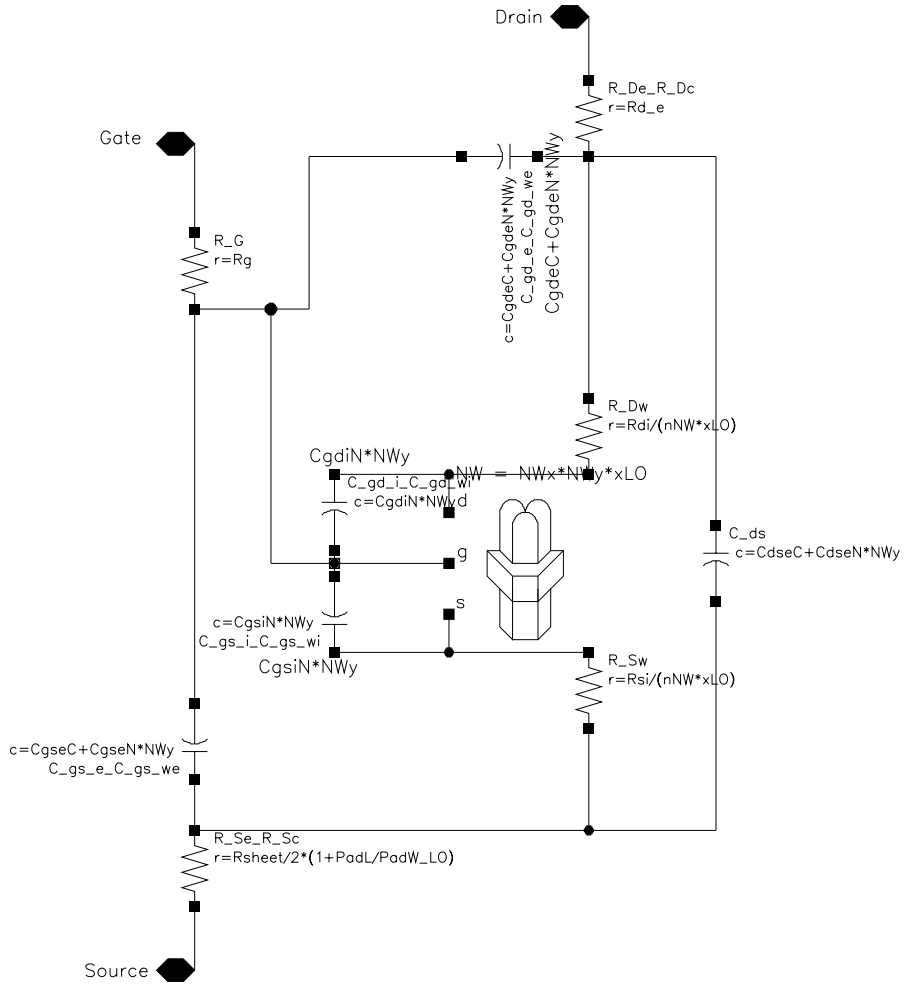

Figure 3: Small signal model of the NWFET in Cadence Virtuoso.

1.2 Millimetre Wave Communication

Millimetre wave (mmWave) is a very broad term that can be classified as the electromagnetic spectrum spanning 30 GHz to 300 GHz, corresponding to a wavelength of 10 mm to 1 mm [8]. In recent years the frequencies around 60 GHz has been of great interest to academia, industry and standardisation bodies and there is already a proposed standard for wireless communication with data rates exceeding 1 Gbps [9].

The decision to use 60 GHz is primarily due to the fact that there exists an unlicensed frequency spectrum in its vicinity. In the U.S. the unlicensed frequency range is 57-64 GHz, for Japan 59-66 GHz and in Europe 57-66 GHz. All together, this provides a common continuous 5 GHz band available around 60 GHz in the major markets [9] (see *figure 4*).

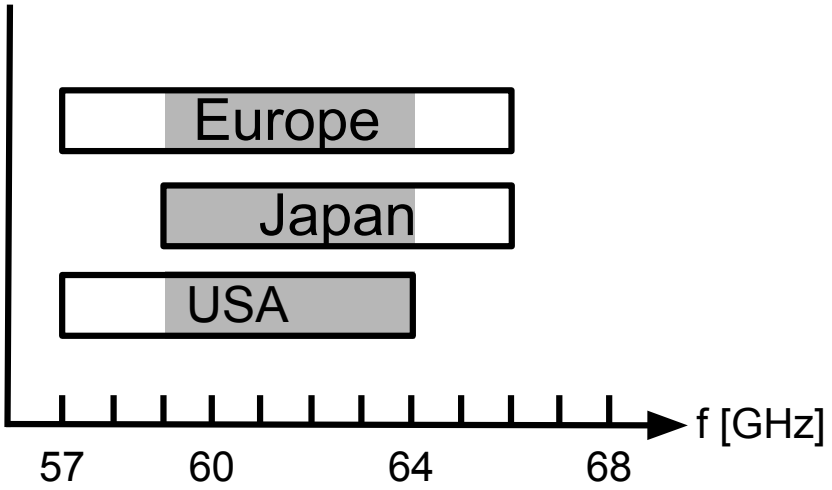

Figure 4: Unlicensed frequencies between 57 and 68 GHz (overlap in grey).

On the down side, increased free space path loss (up to 20 to 40 dB) and atmospheric absorption (15 up to 30 dB/km) for 60 GHz communication systems, combined with reduced multi-path that affects none line-of-sight communications, might make this type of communication unsuitable for longer distances. The high atmospheric absorption is due to the absorption lines of oxygen and can be compared to the path loss of 0.4 dB/km at 94 GHz, a frequency which does not coincide with any absorptions lines [10].

For IEEE 802.16d (the precursor to WiMax[1]), carrier frequencies up to 66 GHz were considered at first, but due to the short range its successor IEEE 802.16e instead chose operating frequencies below 11 GHz. For some applications, limited range might be an attractive feature. For Wireless Personal Area Networks (such as Bluetooth), Multimedia Streaming (e.g. wireless HDMI) and Vehicular Applications, limited range means less interference with adjacent communication systems and increased security [12].

The need for high performance radio frequency components, such as mixers, for frequencies between 57 and 64 GHz is thus of great interest and there are already silicon CMOS implementations for comparison [13–22].

[1]Worldwide Interoperability for Microwave Access [11]

1.3 Frequency Limitations of Silicon CMOS

The frequency capabilities of a transistor is expressed in the form of cut-off frequency (f_T) and maximum frequency (f_{max}). See *table 1* for values for transistors in different technologies.

Definition 1. *At the cut-off frequency f_T, the transistor exhibits unity current gain [23, p. 177]*

Definition 2. *At the maximum frequency of oscillation, f_{max}, the transistor exhibits unity power gain [23, p. 178]*

Ref.	Technology	Node	f_T [GHz]	f_{max} [GHz]	Method
[14]	Silicon CMOS	130 nm	85	135	Measured
[24]	Silicon CMOS	90 nm	140	170	Measured
[25]	Silicon CMOS	65 nm	180	200	Measured
[26]	Silicon On Insulator	90 nm	243	208	Measured
[27]	Silicon On Insulator	45 nm	485	430	Measured
[28]	SiGe BiCMOS	130 nm	240	330	Measured
[5]	III-V NWFET	50 nm	440	380	Simulated
[5]	III-V NWFET	12 nm	1730	1230	Simulated

Table 1: Frequency capabilities for transistors in different technologies.

1.3.1 Silicon On Insulator

Silicon On Insulator (SOI) solves many of the issues with device isolation required for monolithic integration. The traditional junction isolations introduce extra capacitance and there is a risk of current leakage at high enough ambient temperatures. By using a dielectric isolation instead, in the form of a film of crystalline silicon separated by a layer of SiO_2 from the bulk substrate, these difficulties can be overcome. For the past decade, SOI has entered the mainstream of ultra-large scale integration (ULSI) and is needed in order to extend the life of silicon technology. The device scaling made possible by SOI has allowed for silicon circuits operating at frequencies beyond 150 GHz and makes single-cell battery supply (0.5-1 V) feasible [29].

Mixer circuits for radio frequencies of 145-165 GHz [30] have been shown possible to manufacture in 45 nm SOI. A full receiver circuit in 45 nm SOI, capable of ultra-wide band millimetre wave communication with a supply voltage of 0.6 V, has also been reported [31].

1.3.2 SiGe BiCMOS

The high-frequency performance of Heterojunction Bipolar Transistors (HBTs) can be integrated with state-of-the-art CMOS components in a process called BiCMOS. This enables reuse of existing CMOS based cells and low-power applications can be realised. For millimetre wave communication, the choice has been between silicon-germanium (SiGe) HBTs and indium phosphide (InP) HBTs. SiGe HBTs have been preferred for their low cost and adequate performance. The possibility to integrate existing CMOS with high quality HBTs and well-defined passive components makes SiGe BiCMOS a versatile RF technology [32].

Mixer circuits for radio frequencies around 135 GHz [33, 34], 160 GHz [35], 150-170 GHz [36] and 158-165 GHz [37] implemented in BiCMOS have been reported.

2 Theory

2.1 Mixer Fundamentals

The simplest frequency mixer is a nonlinear electronic device that perform a multiplication of two signals in the time domain [23]:

$$(A\cos(\omega_1 t)(B\cos(\omega_2 t)) = \frac{AB}{2}[\cos((\omega_1 - \omega_2)t) + \cos((\omega_1 + \omega_2)t)] \tag{1}$$

Using two different frequencies (ω_1 and ω_2) as the input signal, the resulting output signal contains the sum and difference of those two frequencies. This is practically used in receiver circuits to down-convert a received RF to an intermediate frequency (IF) and in transmitter circuits to up-convert a baseband signal to an appropriate radio frequency for transmission. In both cases the conversion occurs when the applied signal (either RF or baseband) is multiplied with the signal of a local oscillator (LO), see *figure 5*.

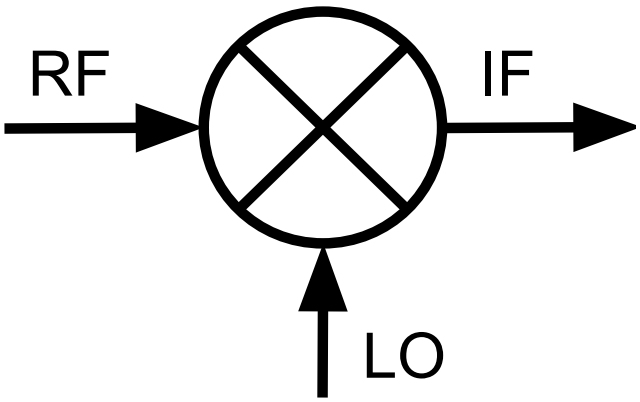

Figure 5: Symbolic representation of an ideal mixer acting as a multiplier.

Apart from the second-order products (1), the output will also contain higher order products on the form:

$$f = mf_1 \pm nf_2 \tag{2}$$

where f is a higher order output frequency and $m, n \in \mathbb{N}$. Note that this will result in f_{LO} when $m = 0$ and f_{IF} when $n = 0$ [38, p. 43].

The third-order product ($m+n = 3$) limits the overall performance and its impact is quantified using the metric IIP3 (2.3.6). The order of intermodulation products are infinite since $m, n \in \mathbb{N}$, but because higher order IPs have smaller amplitudes only second-, third- and fifth-order products ($m + n = 2$, $m + n = 3$ and $m + n = 5$) are of practical importance [38, p. 59].

2.2 Receiver Topologies

2.2.1 Superheterodyne Receivers

The incoming radio frequency is amplified and passed on to a frequency translation section. The frequency translation consists of a frequency mixer and a local oscillator. Heterodyning is the process of converting the RF frequency of the radio signal to a standard intermediate frequency for further processing. The rest of the receiver chain thus only need to deal with the predetermined intermediate frequency, which simplifies the design decisions for filters and other components in the intermediate frequency stage [38, p. 42]. See *figure 6* for details.

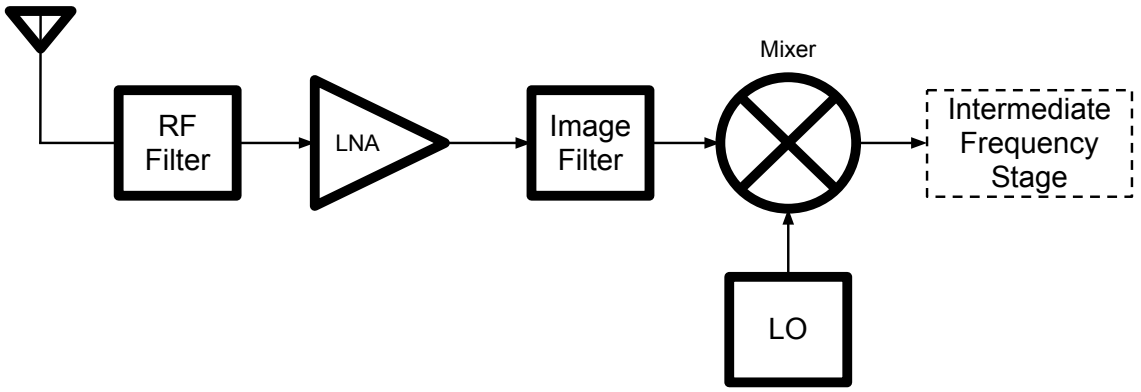

Figure 6: Block diagram of typical superheterodyne receiver front-end.

2.2.2 Homodyne Receivers

The direct frequency conversion (homodyne) receiver has re-emerged and become the de-facto standard for digital cellular systems. The incoming RF signal is amplified and converted directly to baseband in-phase (I) and quadrature (Q) signals [39]. See *figure 7* for details.

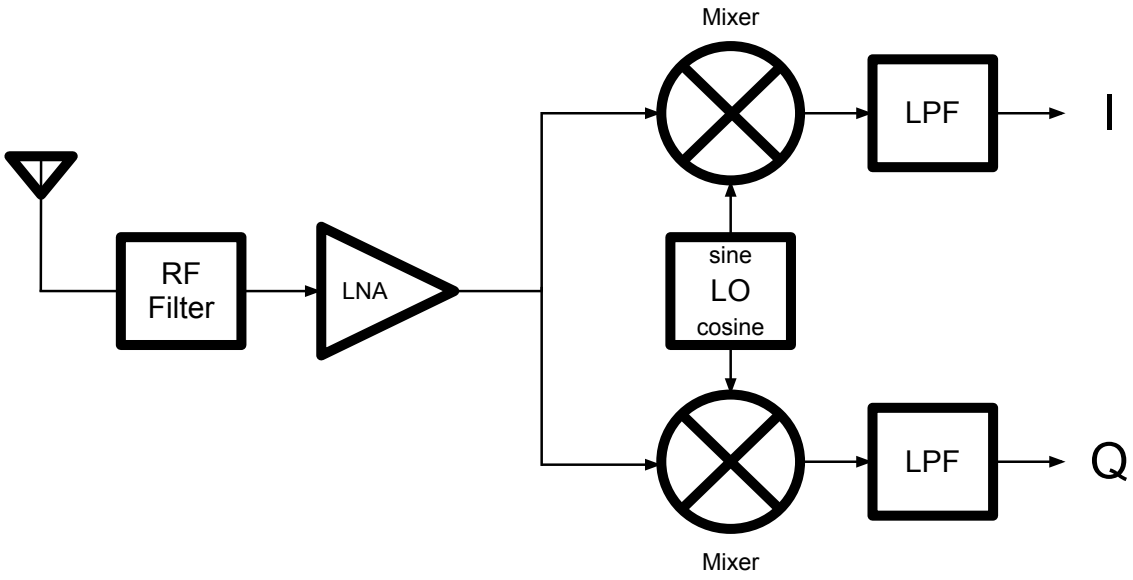

Figure 7: Block diagram of typical homodyne receiver front-end.

2.3 Figures of Merit

2.3.1 Power Consumption

Although power spectrums are possible to simulate [40, p. 16] and measure, the majority of publications (e.g. [13–20, 30, 33, 34, 36, 41–49]) simply state the DC power consumption. The DC current measured at the supply is multiplied with the supply voltage, giving the DC power consumption.

2.3.2 RF to IF Conversion Gain

Definition 3. *Power Conversion Gain [dB]* $= 10 \cdot log\left(\frac{P_{IF}}{P_{RF}}\right)$ *[23, p. 406]*

Definition 4. *Voltage Conversion Gain [dB]* $= 20 \cdot log\left(\frac{V_{IF}}{V_{RF}}\right)$ *[23, p. 406]*

For matched impedance, the voltage conversion gain is equal to the power conversion gain.

The conversion gain is usually measured as function of LO power (with constant RF power, RF and LO frequencies) or as a function of RF frequency (with constant LO power) (e.g [34, 41, 43–45]).

2.3.3 Usable Bandwidth

The usable frequency range for the mixer depends on the requirement for conversion gain. From measurements of voltage conversion gain as a function of RF frequency metrics such as a 3 dB bandwidth can be calculated.

2.3.4 LO to RF and LO to IF Isolation

Isolation between the ports (RF, IF and LO) is important in order to prevent feedthrough (*figure 8*). Since the LO signal is generally much stronger than the RF signal, LO feedthrough to the IF stage might cause problems for subsequent stages. It is also possible for the LO signal (or any of its harmonics) to leak through to the RF stage and be transmitted by the antenna, causing interference with other radio devices [23, p. 409]. The port isolation is expressed as the feedthrough in decibels between two ports.

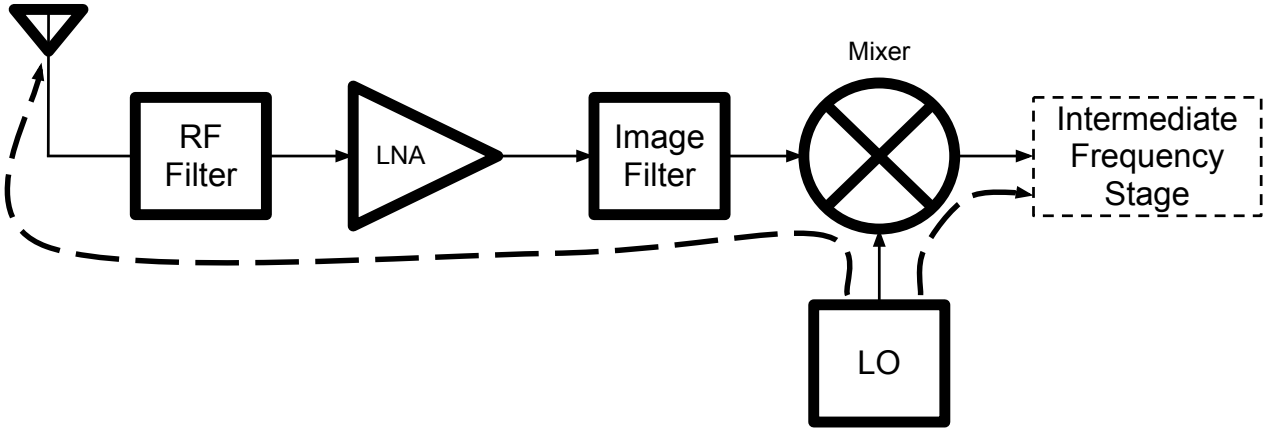

Figure 8: LO to RF and LO to IF feedthrough.

2.3.5 1 dB Compression Point

Ideally, the IF output is proportional to the RF input signal amplitude. For a mixer with a voltage conversion gain, the output amplitude will be equal to the input signal amplitude times the linear gain. However, just as with amplifiers, this relation does not hold true for all input levels. At some point the output power level will deviate from the ideal linear dependence on input power level. The point where the difference between the ideal linear curve and the actual output power curve is 1 dB, is referred to as the 1 dB compression point [23, p. 407-408], see *figure 9*.

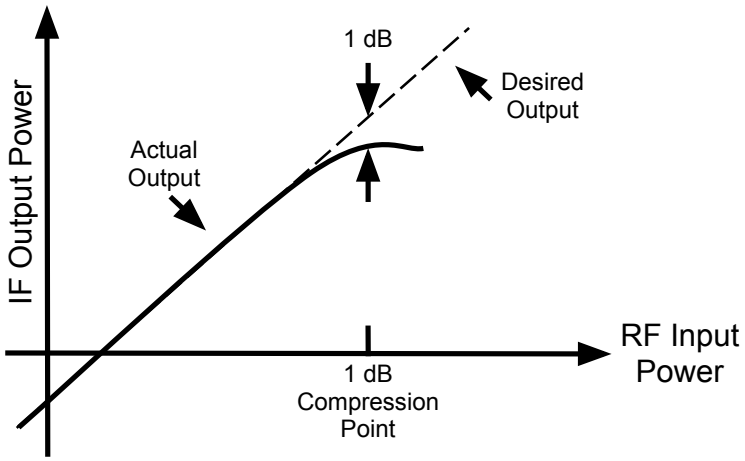

Figure 9: Graphical definition of 1 dB compression point.

2.3.6 Third-Order Intercept Point (IIP3)

Due to the intermodulation effects within the mixer, third-order intermodulation, for instance $2f_{RF1} \pm f_{RF2}$ and $2f_{RF2} \pm f_{RF1}$, will be present. As previously mentioned, the amplitude of the third-order intermodulation is not negligible compared to the desired signal. Therefore it is of interest to know at which point the output level of the third-order intermodulation would be equal to the output level of the desired signal. This point is called the third-order intercept point, see *figure 10*. If the point is referred to using the input power level, it is called IIP3 and if the output power level is used instead, it is called OIP3 [23, p. 408].

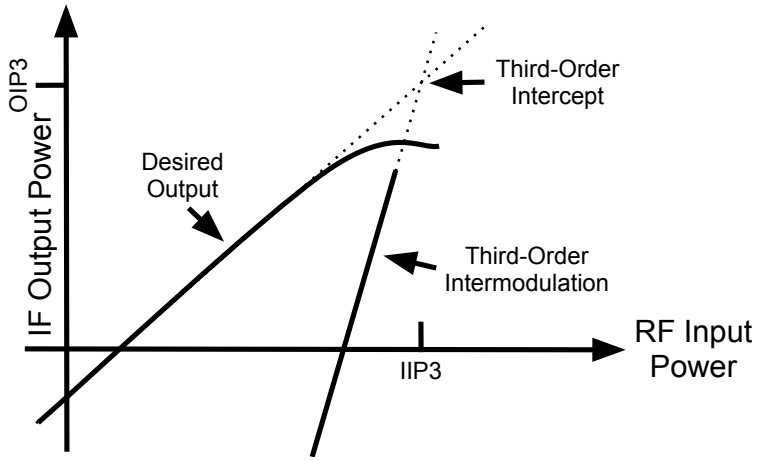

Figure 10: Graphical definition of third-order intercept point.

2.3.7 Noise

The noise of a receiver component is often expressed as a receiver noise ratio [50, p. 124]

$$NR = \frac{(S/N)_i}{(S/N)_o} \tag{3}$$

where $(S/N)_i$ is the signal-to-noise ratio of the input and $(S/N)_o$ the signal-to-noise ratio of the output. The more commonly used noise figure (NF) is the decibel equivalent

$$NF = 10 \cdot log(NR) \tag{4}$$

For a receiver system the overall noise ratio is given by *Friis formula* [50, p. 124]

$$NR = NR_1 + \frac{NR_2 - 1}{A_{p1}} + \frac{NR_3 - 1}{A_{p1} \cdot A_{p2}} + ... + \frac{NR_n - 1}{A_{p1} \cdot A_{p2} \cdot A_{p3} \cdot ... \cdot A_{pn-1}} \tag{5}$$

where NR_1 is the noise ratio of the first stage and A_{p1} is the gain of the first stage.

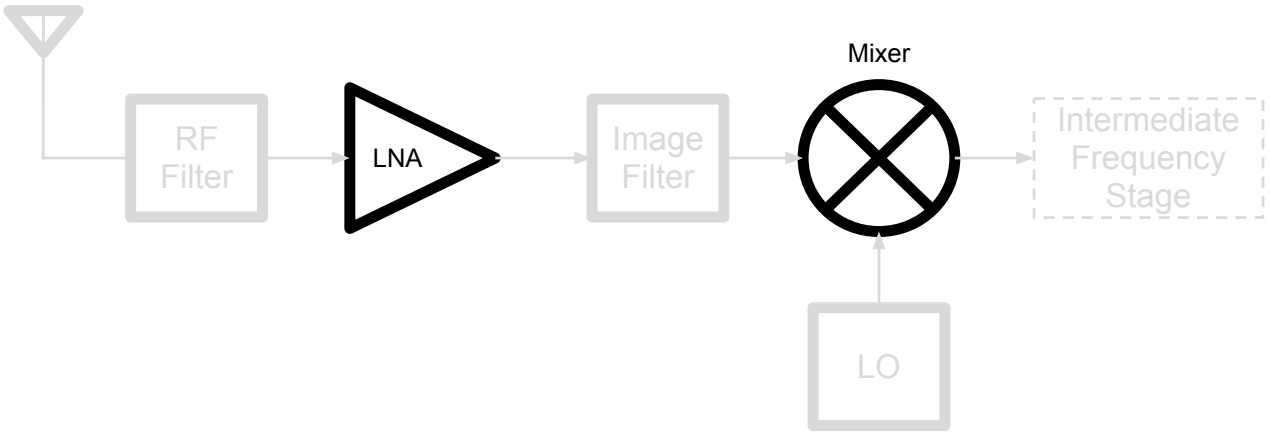

Figure 11: Dominating blocks in the typical receiver.

Consider a receiver system with a LNA with a noise ratio NR_{LNA} and a power gain of $A_{p_{LNA}}$ in series with a mixer with a noise radio NR_{Mixer} and a power gain of $A_{p_{Mixer}}$ (*figure 11*). Then the noise ratio (NR) for the entire system can be expressed as

$$NR_{Receiver} = NR_{LNA} + \frac{NR_{Mixer} - 1}{A_{p_{LNA}}} + \frac{NR_{Rest} - 1}{A_{p_{LNA}} \cdot A_{p_{Mixer}}} + ... \tag{6}$$

thus the LNA is the critical component for overall the noise ratio, whereas the noise ratio of the mixer is suppressed by the gain of the LNA and the gain of the mixer suppresses the noise ratio of subsequent components.

The noise ratio of the mixer itself is the signal-to-noise ratio of the input (RF) divided by the signal-to-noise ratio at the output (IF)

$$NR = \frac{(S/N)_{RF}}{(S/N)_{IF}} \tag{7}$$

For any desired IF, there are two possible frequencies that will yield the same result. One of them is the desired radio frequency (RF) signal and the other is the *image signal*. The two different signals make up one sideband each and are separated by $2f_{IF}$ [23, p. 406].

For this reason, the noise ratio stated can either be dual sideband (DSB) or single sideband (SSB), depending on whether both sidebands contain useful information. Both sidebands will have the same IF noise, but the power level is divided between them. Thus SSB NR will be higher (usually by about 3 dB) than DSB NR. Since DSB NR is numerically smaller, it is usually the figure of merit stated for noise in publications [23, p. 407].

2.4 Combined Figures of Merit

Since there are many different figures of merit to consider for a mixer, it is desirable to be able to combine several figures of merits into a single number and use that as a major performance indicator. Several such combined figures of merits have been proposed

$$FOM_1 = 10 \cdot log(\frac{10^{G/20} \cdot 10^{(IIP3-10)/20}}{10^{NF/10} \cdot P}) \tag{8}$$

The voltage conversion gain G and NF are expressed in dB, while *IIP3* is expressed in dBm. The power consumption P is expressed in mW. The concept is that reduced NF and P are desirable, while maintaining a high voltage conversion gain and *IIP3* [48].

Another combined figure of merit that also takes the radio frequency f_{RF} into account is

$$FOM_2 = 20 \cdot log(f_{RF}) + G - NF + IIP3 - 10 \cdot log(P) \tag{9}$$

The radio frequency f_{RF} is expressed in Hertz, the voltage conversion gain G in dB, the NF in dB, *IIP3* in dBm and the power consumption P in Watts [47].

2.5 Implementation Using Field Effect Transistors

During the first decades of microwave communication, Schottky diodes were the only feasible nonlinear device for radio frequency circuits such as mixers. The diode mixers exhibited a conversion loss of around 6 dB and practical mixers approaching 1 Terahertz had been realised. As monolithic microwave integrated circuits (MMICs) became more common, field-effect transistors where preferred because they provided a conversion gain, rather than loss, and because classical diode mixer designs were impractical to realise in MMIC [51, p. 6-10].

2.5.1 Active and Passive Mixers

Using field-effect transistors it is possible to realise both active and passive mixers. A comparison between the advantages and disadvantages of active and passive mixers is shown below:

	Advantages	**Disadvantages**
Balanced Active	• Provide gain • Good LO to RF isolation • Requires low LO drive	• Higher NF than passive mixers • Lower linearity than passive • Power consumption
Passive	• Good linearity • Low NF	• Poor LO to RF isolation
Balanced Passive	• Even better linearity • Reasonably low NF • Good LO to RF isolation	• Requires strong LO drive • Requires off-chip baluns

Table 2: Comparison between active and passive mixers [52, p. 73-90].

This project focuses solely on balanced active mixers. For a detailed outline of design of passive radio frequency mixers using nanowire field-effect transistors, see the thesis report *Design and Simulation of Passive Nanowire-Based Mixers* by Mohammad Khorramnejadi, presented at the Department of Electrical and Information Technology at Lund University, Sweden in 2012.

2.6 Different Active Mixer Topologies

2.6.1 Single Device (Square-Law Mixer)

Consider a MOSFET in saturation

$$I \propto V^2 \tag{10}$$

Applying two periodical signals

$$I \propto (A \cdot \cos(\omega_1 t) + B \cdot \cos(\omega_2 t))^2 \tag{11}$$

which expands to

$$A^2 \cdot \cos^2(\omega_1 t) + 2AB \cdot \cos(\omega_1 t) \cdot \cos(\omega_2 t) + B^2 \cdot \cos^2(\omega_2 t) \tag{12}$$

Using the trigonometric identity

$$\cos(\theta) \cdot \cos(\varphi) = \frac{\cos(\theta - \varphi) + \cos(\theta + \varphi)}{2} \tag{13}$$

the final result contains the multiplication of the two periodical signals in the time domain

$$A^2 \cdot \cos^2(\omega_1 t) + \boldsymbol{AB \cdot \cos(\omega_1 t + \omega_2 t) \cdot \cos(\omega_1 t - \omega_2 t)} + B^2 \cdot \cos^2(\omega_2 t) \tag{14}$$

Thus it is possible to realise a mixer using only a single MOSFET.

Even though single device mixers are uncommon for practical usage, at least one single device mixer for millimetre-wave communication has been reported. Low-power operation was achieved by choosing a biasing point close to the threshold voltage. The major practical implementation problem mentioned is the need for a hybrid or elaborate power combining circuit to combine the LO and RF signals [14].

An inherit drawback for the single-device mixer is that the isolation between the LO and RF signal is poor, since they are applied in series [23, p. 414].

2.6.2 Single-Balanced

If the LO input voltage (V_{LO}) is high enough, the periodical LO signal will cause alternating switching of transistor M1 and M2. The alternating switching will bring the tail current (I_D of transistor M3) back and forth between the two IF output nodes [23, p. 417] (*figure 12*).

Since there is no direct signal path from LO to RF (as in the case with the single device mixer in 2.6.1), the port isolation is ideally infinite. However, due to the parasitic capacitance between gate and drain, leakage does occur [52, p. 77].

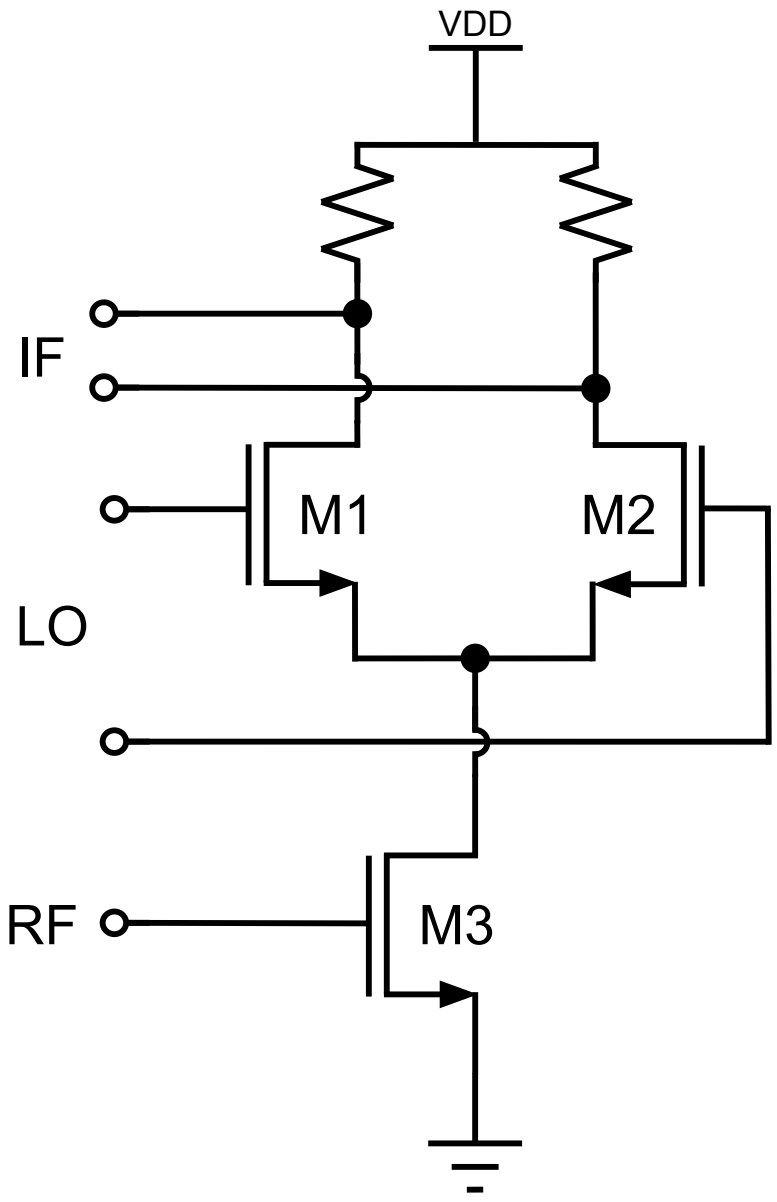

Figure 12: Simplified circuit diagram of a single-balanced mixer.

The different stages of the single balanced mixers need to be considered in order to derive the voltage conversion gain, see *figure 13*.

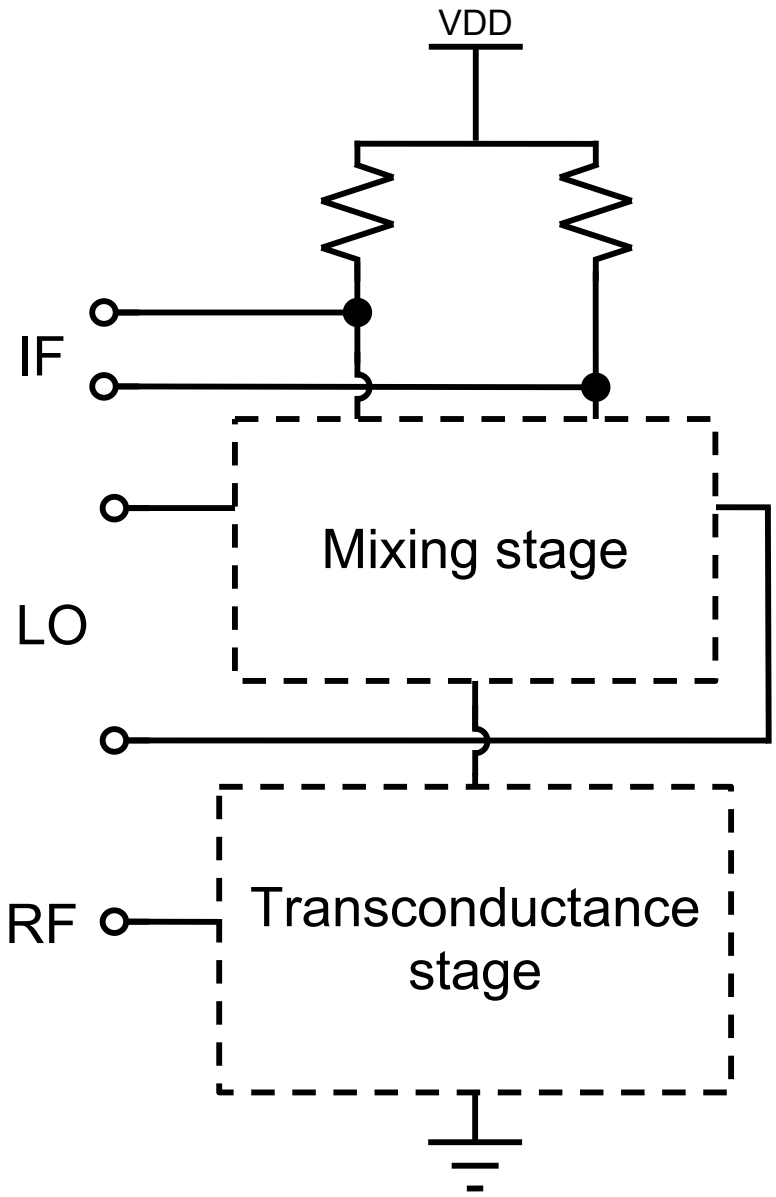

Figure 13: The different stages of the single-balanced mixer.

The conversion gain depends on three parts: the transconductance of the RF input transistor ($g_{m,rf}$), the switching gain or loss at the mixing stage (A_{sw}) and the output impedance (R_0) [52, p. 77]

$$A_v = g_{m,rf} \cdot R_o \cdot A_{sw} \tag{15}$$

The switching gain (A_{sw}) is a function of the LO drive and the overdrive voltage of the transistor pair making up the mixing stage. It will be proportional to the LO amplitude as long as the LO amplitude is smaller than the overdrive voltage. In turn, the overdrive voltage will depend on the drain current of the transistor in the transconductance stage [52, p. 77].

2.6.3 Double-Balanced (Gilbert Cell)

The Gilbert cell was first presented in Barrie Gilbert's landmark paper [53] and has since been the topology of choice for many mixer designs (see *figure 14*). In essence it consists of two single-balanced mixers that couples differential LO signals into the same IF output. At the mixer stage, the IF output is now connected to two switching transistors with π radians phase difference between their respective LO signals, causing the LO leakage to cancel out [52, p. 77-78]. This topology therefore exhibit better port isolation than the single-balanced (2.6.2) and single-device mixer (2.6.1).

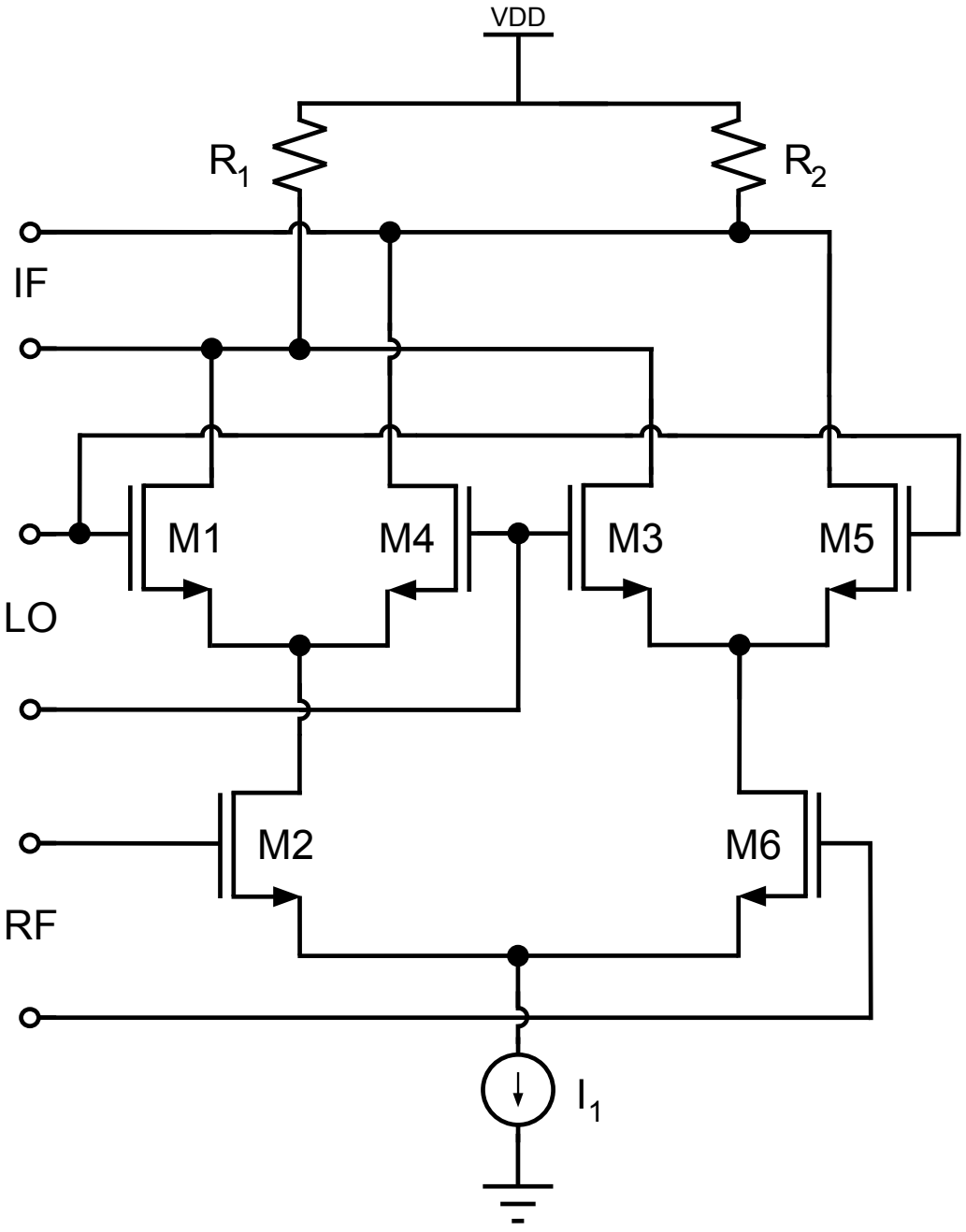

Figure 14: Simplified circuit diagram of the Gilbert cell.

The apparent drawback of this topology is the increased number of transistors needed, which requires more chip surface area and makes it more sensitive to process variations.

The double-balanced mixer will exhibit half of the voltage conversion gain as in the single-balanced case since ($g_{m,rf}$) and (A_{sw}) will remain the same, but for the same supply voltage the load resistance R_0 has to be lowered due to the limited voltage headroom [54, p. 377]. As seen in *figure 15*, the stages are the same as for the single-balanced mixer.

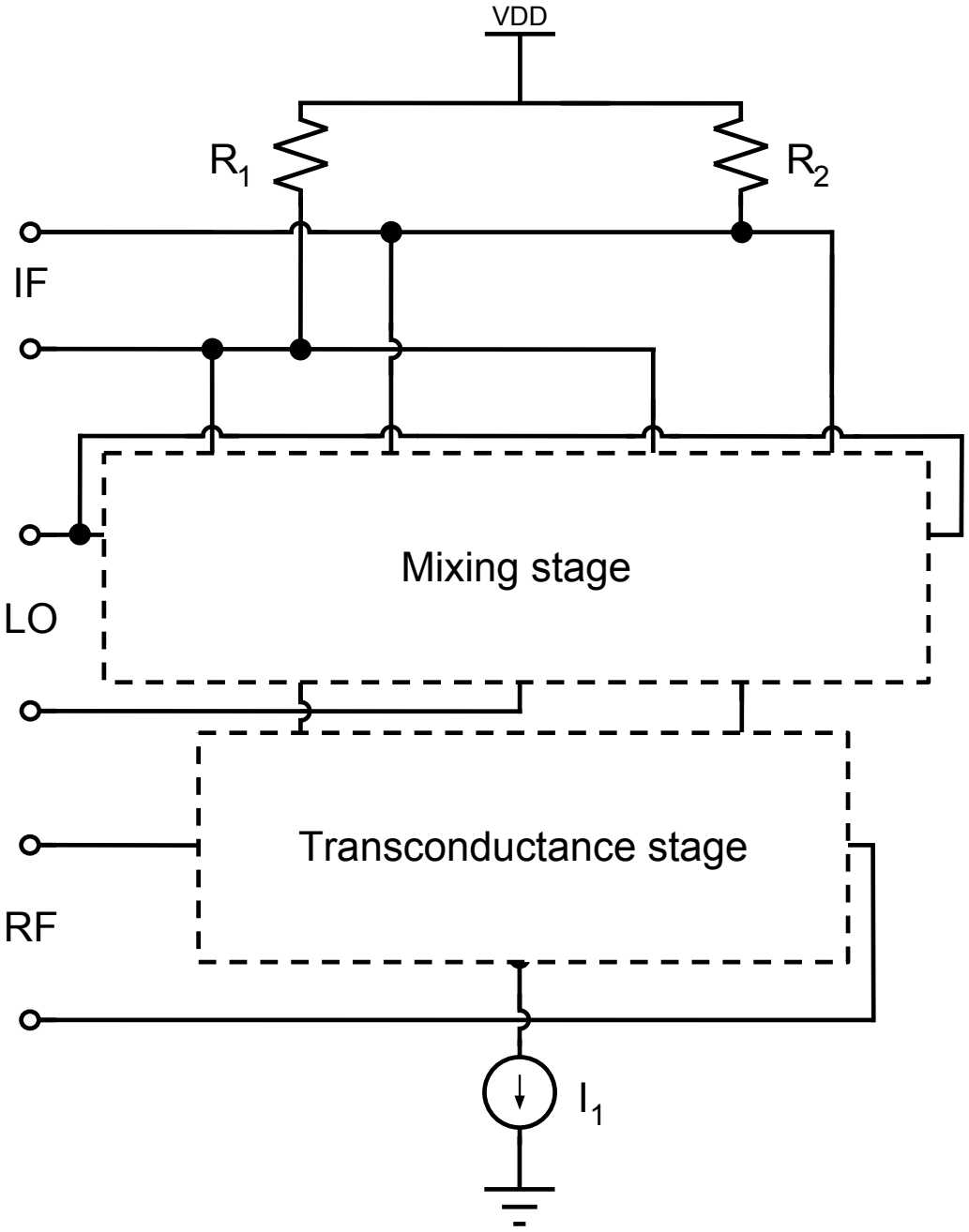

Figure 15: The different stages of the gilbert cell.

For a known $g_{m,rf}$ and R_0 with an LO signal in the form of a square wave, the voltage conversion gain (A_v) can be estimated as [54, p. 390]

$$A_v = \frac{2}{\pi} g_{m,rf} R_0 \tag{16}$$

NF_{SSB} referred to a 50 Ω source (R_S) can be estimated as [54, p. 390]

$$NF_{SSB} = 1 + \frac{\pi^2 kT \left(\frac{\gamma}{g_{m,rf}} + \frac{2}{g_{m,rf}^2 R_0} \right)}{4kTR_S} + 3 \tag{17}$$

The last term 3 is due to the noise figure of the double-balanced mixer being 3 dB higher than an equivalent single-balanced mixer (same transistor sizing and supply voltage) [54, p. 379]. This noise estimation is known to be off by as much as 4 dB, due to an oversimplification not taking contribution of the switches and aliasing of white noise by the transconductance stage into account [55].

The overall noise figure can be improved at the cost of higher power consumption by scaling the entire mixer with a factor α, which will lower the input referred noise voltage $\sqrt{\overline{V_{n,in}^2}}$ by a factor $\sqrt{\alpha}$ [54, p. 384], see *figure 16*.

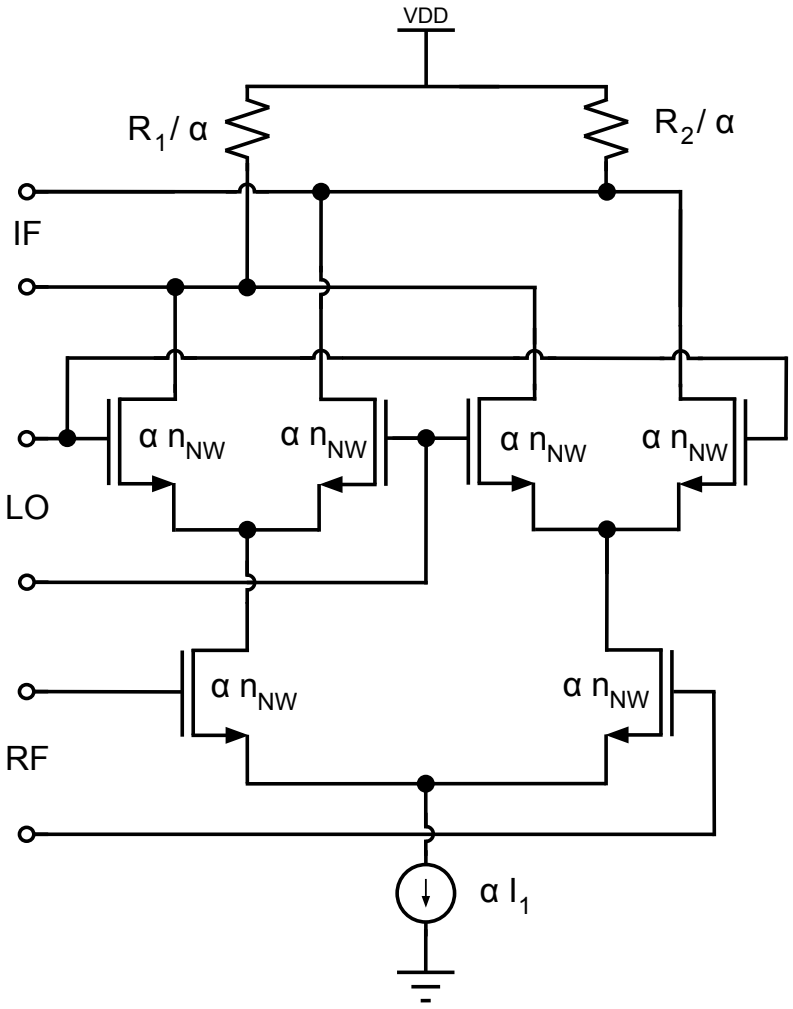

Figure 16: Scaling of entire mixer to reduce noise.

2.7 Performance Enhancements

2.7.1 Source Degeneration

The RF stage of the single- and double-balanced mixer is essentially a common-source amplifier stage. If a degeneration resistor (R_S) is placed in series with the source terminal (see *figure 17*), as the input voltage (V_{in}) increases, so does the drain current (I_D) and the voltage drop over R_S.

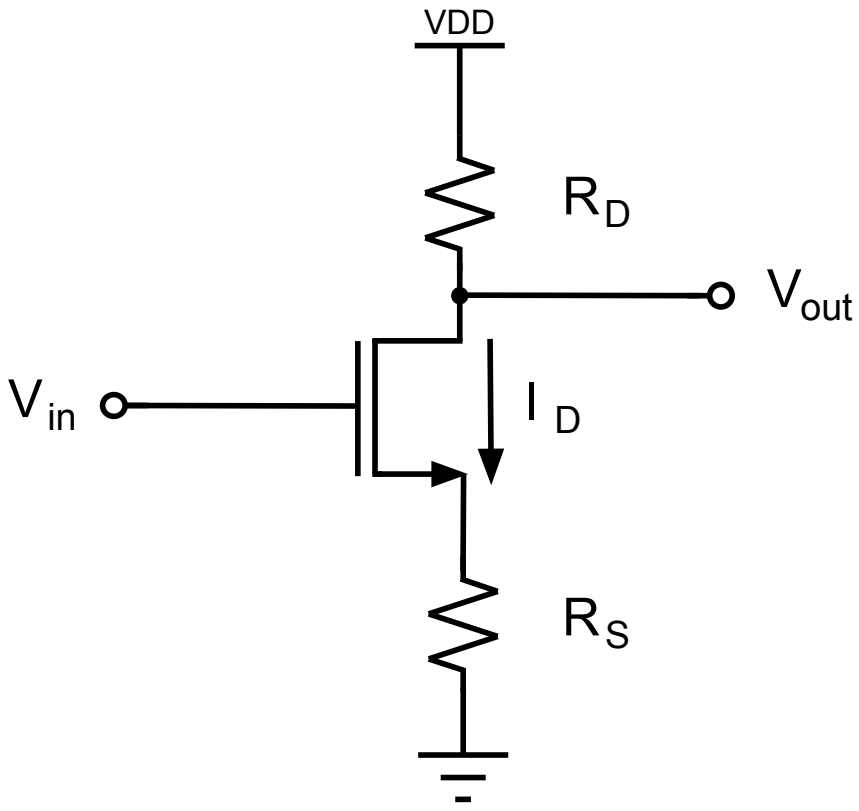

Figure 17: Common-source amplifier with resistive source degeneration.

This leads to a smoother variation of I_D, since part of V_{in} now appears across R_S rather than as the gate-source overdrive. The total gain of the amplifier (G_m) becomes a weaker function of the transconductance of the transistor (g_m) [56, p. 60-62]

$$G_m = \frac{g_m}{1 + g_m R_S} \tag{18}$$

If $R_S \gg 1/g_m$

$$G_m \approx \frac{1}{R_S} \Rightarrow \Delta I_D \approx \frac{\Delta V_{in}}{R_S} \tag{19}$$

The resulting linearisation comes with the cost of lower gain and higher noise [56, p. 61]. Source degeneration at the RF stage can thus increase linearity of the mixer, but will lower the conversion gain and increase the noise figure.

If reactive components are considered for source degeneration, it can be shown, using Volterra series expansion, that the magnitude of the third-order intermodulation point ($|IM_3|$) for the transconductance stage of a single-balanced mixer depends on [57]

$$|1 + j\omega C_{gs}[Z_s(\omega_1, L_s) + Z_g(\omega_1, L_s)]| \tag{20}$$

where Z_g is the impedance at the gate of the transistor, Z_s is the impedance at the source of the transistor and C_{gs} is the gate-source capacitance.

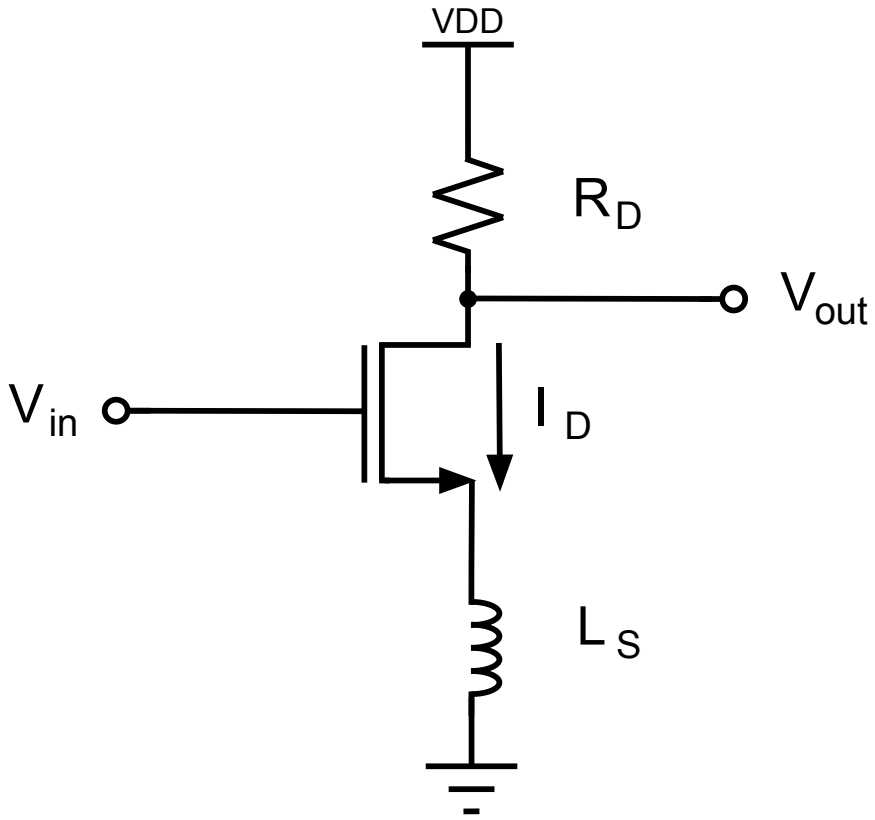

Figure 18: Common-source amplifier with inductive source degeneration.

With inductive degeneration at the source (see *figure 18*), $j\omega C_{gs} Z_s(\omega_1, L_s)$ is a negative real number that can partially cancel out the first term, thus lowering the magnitude. This sort of cancellation cannot be achieved using resistive nor capacitive source degeneration [57].

Overall, inductive degeneration shows better linearity performance than resistive and capacitive, but the bandwidth of the circuit might be narrowed [58].

2.7.2 G_m-boosting

From (15) it is clear that the conversion gain of the mixer will depend on the order of amplification at the RF input stage. By amplifying the RF voltage before the RF transistor stage (see *figure 19* and *figure 20*), the overall conversion gain can be increased [59]

$$I_{RF} = I_D - (1 + A)g_m \cdot v_{RF}(t) \tag{21}$$

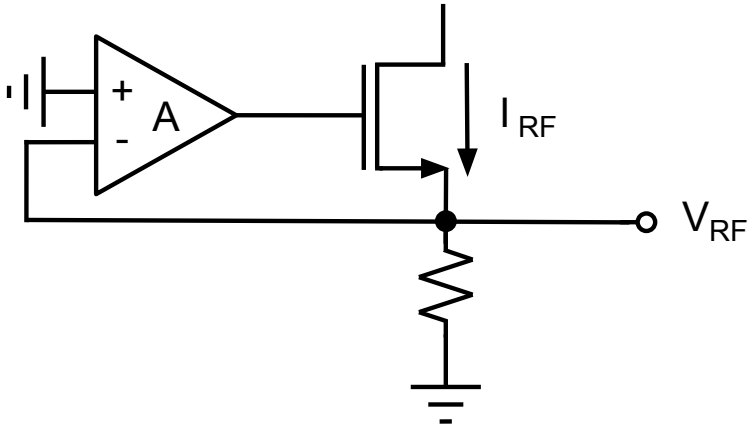

Figure 19: General G_m-boosting at the RF stage.

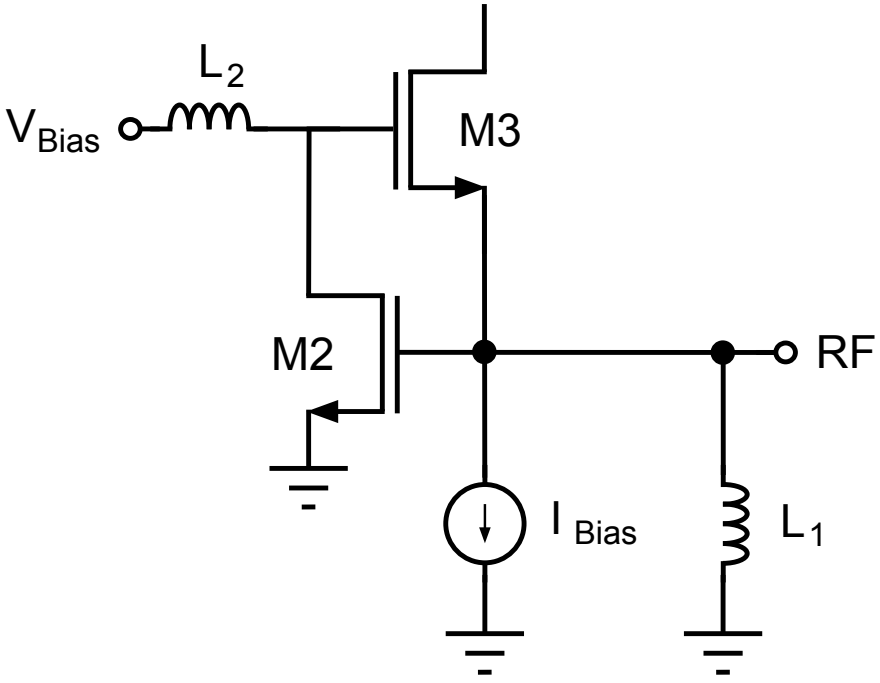

Figure 20: Detailed G_m-boosting at the RF stage, as outlined in [59].

It is also possible to utilise the same technique at the mixing stage, in order to relax the LO power level requirement of the mixer. A 60 GHz down-converting mixer with a conversion gain of 12 dB at -13 dBm LO power has been presented [60].

2.7.3 Current Bleeding

Linearity is often limited by the transconductance stage (the RF input transistor). For a common source MOSFET with an α taking channel velocity saturation and mobility degradation in to account, IIV_3 (the voltage equivalent of *IIP3*) can be expressed as [52, p. 82]

$$IIV_3 = \sqrt{\frac{8}{3}\frac{1}{\alpha}V_{od}\left(1+\frac{1}{2}\alpha V_{od}\right)(1+\alpha V_{od})^2} \tag{22}$$

Since IIV_3 increases with the overdrive voltage V_{od} and the subsequent mixing stage does not add much distortion, a common technique for increasing linearity of a Gilbert mixer without sacrificing conversion gain is to introduce *current bleeding* (also known in the literature as *current stealing* and *charge-injection*) at the RF input transistor.

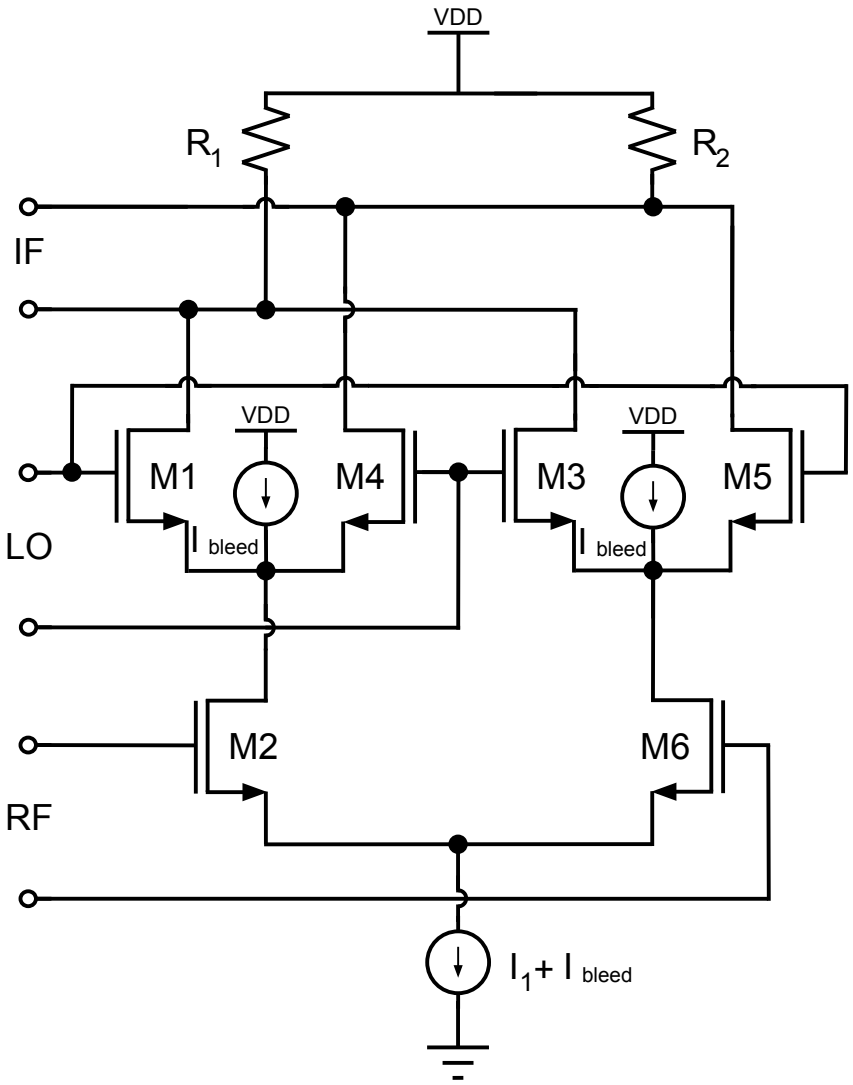

Figure 21: Double-balanced Gilbert cell with current bleeding.

The goal is to increase the drain current of the RF input transistor and then divert the unnecessary DC current away from the signal path (see *figure 21*) [52, p. 82]. Various ways to implement current bleeding have been proposed [16, 61, 62].

2.8 SpectreRF Simulations in Cadence Virtuoso

2.8.1 Periodic Steady-State Analysis (PSS)

PSS analysis computes the periodic steady-state response of a circuit at a specified fundamental frequency by first performing a transient phase to initialise the circuit. When the circuit is initialised a technique called the shooting method is used to compute the periodic steady-state solution for the circuit. The shooting method usually requires about five iterations on most circuits. Thanks to Cadence's Fourier integral method, the accuracy approaches that of harmonic balance simulators for near-linear circuits, and offers even greater accuracy for strongly nonlinear circuits [63, p. 32].

2.8.2 Periodic AC Analysis (PAC)

PAC analysis is a small-signal analysis used to compute transfer functions for circuits that exhibit frequency translation, such as mixers. It must be performed after a large signal PSS analysis. Depending on whether the mixer is down-converting or up-converting, the transfer function or sideband is labeled -1 and +1 respectively [63, p. 44]

2.8.3 Quasi-Periodic Steady-State Analysis (QPSS)

The quasi-periodic steady-state (QPSS) analysis allows for the computation of the response to several moderately large input signals in addition to a strongly nonlinear tone which represents the LO signal. It is assumed that the circuit responds in a strongly nonlinear fashion to the large tone and in a weakly non-linear fashion to the moderate tones. QPSS can be used to model intermodulation distortion [63, p. 71].

2.8.4 Quasi-Periodic AC Analysis (QPAC)

The quasi-periodic AC (QPAC) analysis computes transfer functions for circuits that exhibit multi-tone frequency translation, such as mixers. QPAC is similar to the conventional AC analysis, but instead of linearise about a simple DC operating point, the circuit is linearised about a quasi-periodically time-varying operating point. This produces transfer-functions that include frequency translation [63, p. 87].

2.8.5 Periodic Transfer Function Analysis (PXF)

PXF analysis computes the transfer functions from any source at any frequency to a single output at a single frequency. It is used to compute quantities such as conversion efficiency, image and sideband rejection, and LO feed through. Just like the PAC analysis, a PXF analysis must follow a PSS analysis [63, p. 56-57].

2.8.6 Periodic Noise Analysis (Pnoise)

Periodic Noise analysis (Pnoise) is capable of modelling frequency conversion effects of the circuit noise. It is therefore suitable for computing the noise behaviour of mixers. It must be performed after a large signal PSS analysis. By default the noise figure will be single sideband (SSB) [63, p. 61].

2.9 MOS Transistor Noise

2.9.1 Shot Noise

The passage of carriers across a junction can be modelled as a random event, thus the total current is composed of a large number of random independent current pulses. This noise is called shot noise [64, p. 736]. The gate leakage current I_G causes shot noise in the MOS transistor [64, p. 747].

2.9.2 Thermal Noise

The channel of the MOS transistor is resistive and thus exhibits thermal noise, due to random thermal motion of the carriers [64]. For simplified modelling the thermal noise is approximated as white noise, which means the noise power spectrum is constant throughout the frequency spectrum and only depends on the absolute temperature.

2.9.3 Flicker Noise

Another noise component is flicker noise, which exhibits a frequency dependence $1/f$. Flicker noise originates mainly from traps associated with contamination and crystal defects. Because of this dependence, flicker noise dominates at lower frequencies [64, p. 741], but due to up-conversion it also has a serious impact on RF circuits [65].

2.9.4 Noise Spectral Density

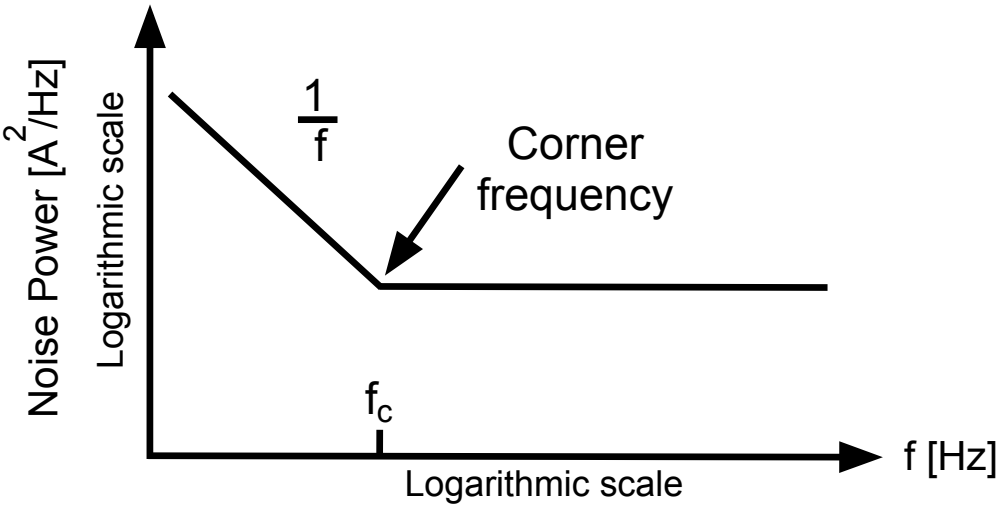

Figure 22: Typical equivalent noise voltage spectral density for a MOSFET [64, p. 763].

In a noise spectrum, as seen in *figure 22*, the flicker noise dominates up to the corner frequency, after which the frequency independent thermal noise will be much larger than the flicker noise.

Experimentally it has been found that the flicker noise in the MOS transistor can be represented by a drain-source current generator that can also incorporate the thermal noise [64, p. 747]

$$\overline{i_d^2} = \underbrace{4kT\gamma g_m \Delta f}_{\text{Thermal noise}} + \underbrace{K\frac{I_D^a}{f}\Delta f}_{\text{Flicker noise}} \tag{23}$$

where k is the Boltzmann constant, T is the absolute temperature in Kelvin, γ is the excess noise coefficient ($\frac{2}{3}$ for long-channel devices and even as high as 2 for short-channel) [54, p. 43], g_m is the device transconductance at the operating point, I_D is the drain bias current, K is a device dependent constant and a is a constant between 0.5 and 2.

The shot noise due to gate leakage is given by [64, p. 747]

$$\overline{i_g^2} = 2qI_G\Delta f \tag{24}$$

where q is the elementary charge and I_G is the gate leakage current.

The previously mentioned noise sources are independent of each other, but due to the capacitance between the gate and channel there is a gate noise current source coupled with the thermal noise term in (23) [64, p. 747]

$$\overline{i_g^2} = \frac{16}{15}kT\omega^2 C_{gs}^2 \Delta f \tag{25}$$

where $C_gs = (2/3)C_{ox}WL$. This correlated noise has been proven to be challenging to implement in compact modelling languages such as Verilog-A [66].

2.9.5 Modelling Correlated Noise

The PSP model, a compact MOSFET model, jointly developed by NXP Semiconductors (formerly part of Philips) and Arizona State University (formerly at The Pennsylvania State University) contains a correlated noise model . The model is widely accepted and in December 2005, the Compact Model Council (CMC) elected PSP as the new industry standard for compact MOSFET modelling. A Verilog-A implementation of the PSP model can be found in its entirety at the PSP model web site `http://pspmodel.asu.edu` and the NXP Semiconductors web site `http://www.nxp.com/models` [67].

Modelling frequency-dependent noise spectral density directly in Verilog-A has proven to be limited and in order to model the correlation between the noise sources additional internal nodes have to be added [67, p. 99]. It has been shown that correlated noise can be modelled successfully using Verilog-A [66]. The noise aspects of the PSP model should provide a sufficient starting point for extending the virtual-source model of the III-V NWFET to incorporate a more comprehensive noise model that can also take the correlation between the gate-current and the drain-source noise into account.

3 Method

3.1 Automated RF Mixer Test Bench in Cadence Virtuoso

Using the Open Command Environment for Analysis (OCEAN) in Cadence Virtuoso a fully automated test flow was created to evaluate the performance of mixer circuits during the design phase. The test flow consists of several modules (see *figure 23*) that can be run independent of each other and the final data is analysed in MatLab in order to produce plots and summaries. The schematic for the test bench can be found in the appendix (*Test Bench Implementation*) along with the SKILL and MatLab code (*Code*).

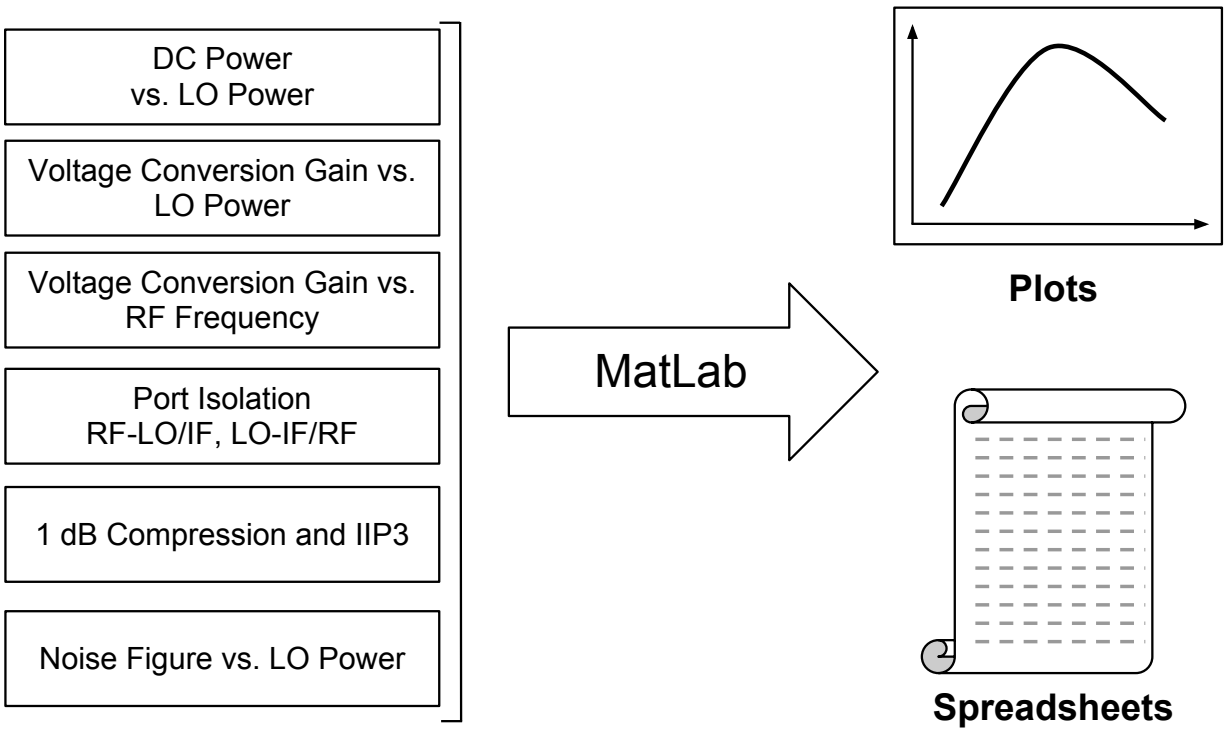

Figure 23: Overview of test modules and workflow.

The test bench (*figure 6*) consists of three signal sources: Local Oscillator (LO), Radio Frequency (RF) and Intermediate Frequency (IF). The RF signal can either be applied to the mixer as a balanced or unbalanced signal, depending on whether the ideal balun is used or not. The LO signal is converted to a balanced signal. The most common configuration during testing is to keep the RF and IF sources operating in DC mode and use a sine wave as the LO signal. The signal source configuration is done automatically by the automated test flow.

The tests were implemented based on a manual published by Cadence Systems [40], the company behind Cadence Virtuoso. The manual is commonly cited and reused for lab manuals in university radio electronics courses. Some adaptations had to be made since the manual focuses on low-bandwidth mixers in the megahertz regime.

3.1.1 DC Power Consumption versus LO Signal Power

Runs a Periodic Steady State (PSS) analysis for the centre frequency of the desired frequency range and does a linear sweep of the LO signal power (*pLO*) [40, p. 9]. Using a virtual ammeter (an iprobe from the analogue library) connected between the the supply pin (VDD) and the voltage source, the average DC current is calculated. The supply voltage is determined by a design variable (*vdd*). By multiplying the time average DC current with the supply voltage, the average DC power consumption during operation is determined. The result is plotted in MatLab as a function of LO signal power level (*pLO*).

3.1.2 Voltage Conversion Gain versus LO Signal Power

Runs a Periodic Steady Steady (PSS) analysis for the centre frequency and does a linear sweep of the LO signal power (*pLO*) followed by a Periodic AC (PAC) analysis for the same frequency. The voltage conversion gain is the logarithmic peak voltage of the mixer output signal (*IF*). The result is plotted in MatLab as a function of LO signal power level (*pLO*) [40, p. 9].

3.1.3 Voltage Conversion Gain versus RF Frequency

Identical to *Voltage Conversion Gain versus LO Signal Power*, but sweeps the entire desired radio frequency spectrum in order to get a the voltage conversion gain at different radio frequencies (*RF*). The result is plotted in MatLab as a function of radio frequency (*RF*).

3.1.4 Port Isolation

Runs a Periodic Steady State (PSS) analysis for the centre frequency followed by a Periodic AC (PAC) analysis and a Periodic Transfer Function (PXF) analysis in order to get the transfer functions between the different ports (*LO*, *RF* and *IF*) [40, p. 28].

The logarithmic peak voltage transfer between the ports is saved for *RF* to *LO*, *RF* to *IF*, *LO* to *IF* and *LO* to *RF*. The average values for the port transfer functions are calculated in MatLab.

3.1.5 1 dB Compression and Third-Order Intercept

Runs a Quasi-Periodic Steady State (QPSS) analysis for the fundamental tones (consisting of f_{LO} and f_{RF}) and sweeps the RF signal power (*pRF*) from -70 dBm to +5 by default. The QPSS is followed by a Quasi-Periodic AC (QPAC) analysis [40, p. 25].

The resulting 1 dB compression curve and third-order intercept curve are exported and analysed in MatLab where the 1 dB compression point and output/input referred third-order intercept (OIP3 and IIP3). Since the calculations depend on a linear regression compared against the measured results it is important to do an extensive sweep of the radio frequency power (*pRF*) during the QPSS analysis (preferably more than 20 steps).

3.1.6 Noise Figure versus LO Signal Power

Runs a Periodic Steady Steady (PSS) analysis for different LO power and uses the results to run a Periodic Noise (Pnoise) analysis for an intermediate frequency of 100 MHz.

3.2 Implementation of Noise Model for the NWFET

The underlying model of the III-V NWFET is implemented using the hardware description language Verilog and the component used in the Cadence Virtuoso is a symbol representation of a cell-view with a comprehensive small-signal model [5] (*figure 2*). This will add noise in the form of thermal noise from the resistors in the small-signal model, see *figure 24*.

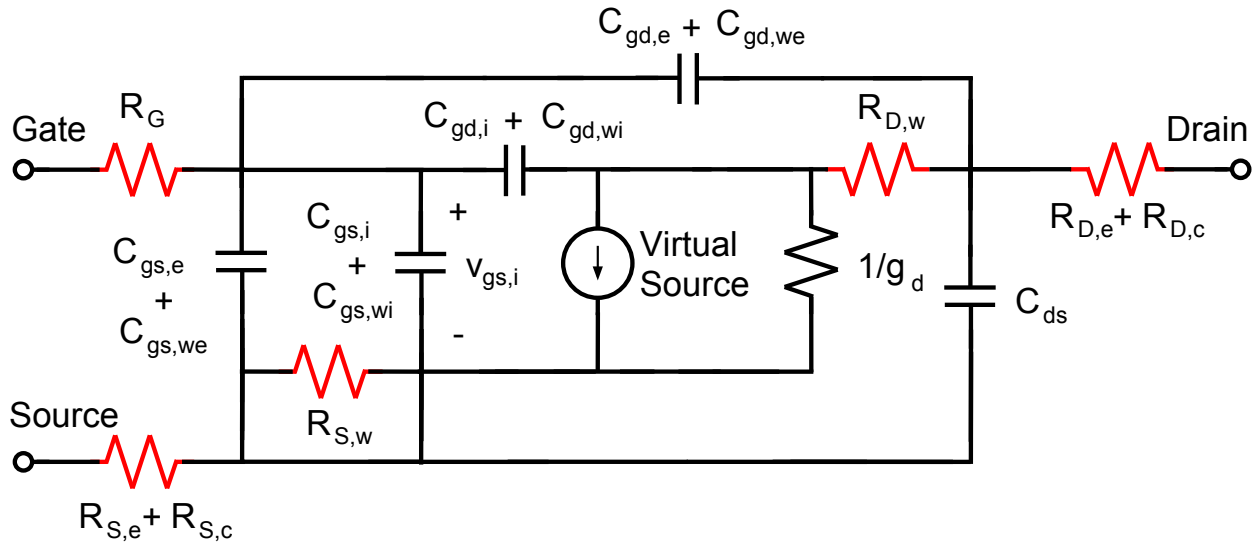

Figure 24: Source of noise in the original small signal model of the NWFET.

In order to evaluate different mixer configurations, an improved noise model for the transistor has to be considered. Since the original III-V NWFET model does not contain any noise sources for the channel, the compact model in Verilog has to be updated to produce white- and flicker noise.

The standard way of implementing noise in Verilog is using the built-in functions:

white_noise function
Generate white noise, noise whose current value is completely uncorrelated with any previous or future values [68, p. 117].

```
white_noise( power [ , "name"])
```

where *power* is the power of the source and *name* is a label for the noise source.

flicker_noise function
Generate pink noise that varies in proportion to $\frac{1}{f^{exp}}$ [68, p. 117].

```
flicker_noise( power, exp [ , "name"])
```

where *power* is the power of the source at 1 Hz and *name* is a label for the noise source.

By assuming no gate leakage, the noise source can be simplified to a drain-source noise source corresponding to (23)

$$\overline{i_d^2} = \underbrace{4kT\gamma\left(\frac{I_{DS}}{V_{DS}}\right)\Delta f}_{\text{Thermal noise}} + \underbrace{nWf(V_{GS}, V_{DS_{sat}})\Delta f}_{\text{Flicker noise}} \tag{26}$$

where γ is assumed to be 2 (short-channel devices), n is the number of wires in the array, W is the circumference of a single wire (in μm) and $f(V_{GS}, V_{DS_{sat}})$ is fitted from experimental noise measurements by Persson et. al, see *figure 25*.

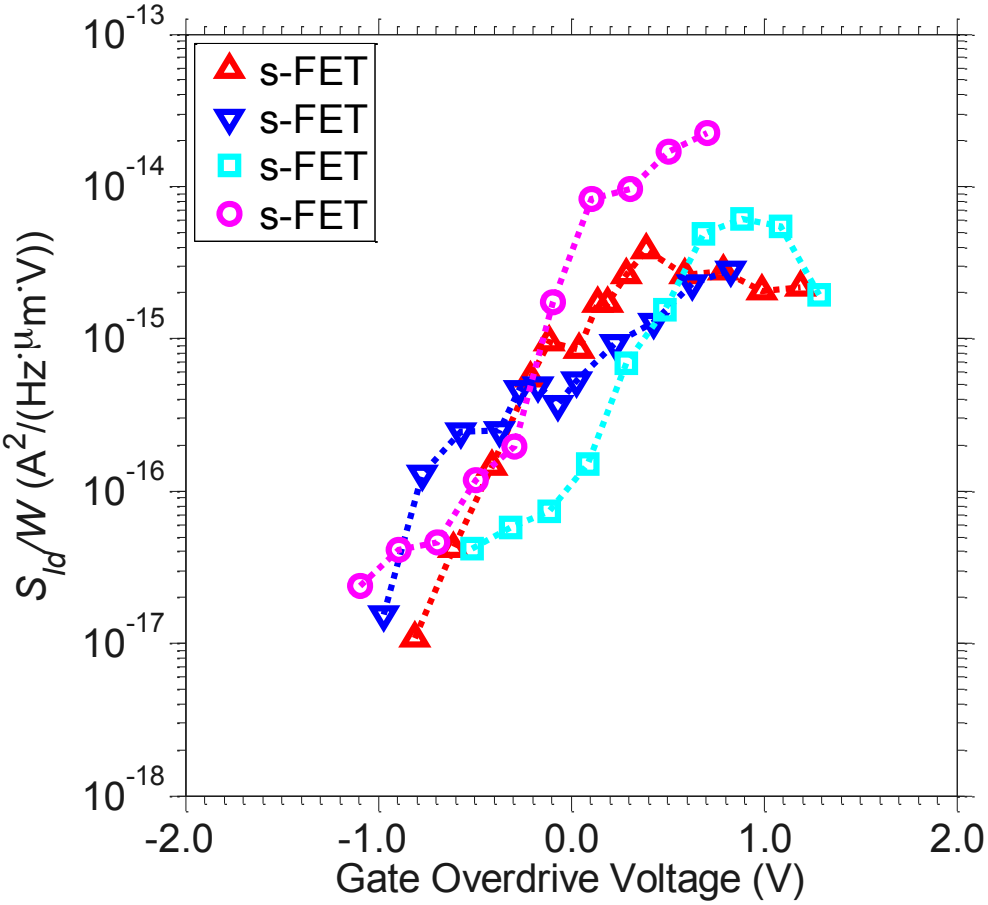

Figure 25: Measured noise data for the NWFET, courtesy of Karl-Magnus Persson.

The measured data is from four different single wire devices (s-FET), but it is assumed that the noise will scale linearly with the number of wires in an array based NWFET. It is also assumed that the noise outside of the measured gate overdrive voltage range of -1.0 V to 1.0 V follows the same behaviour and can thus be extrapolated.

A linear regression was done on the above data in order to find a suitable expression for $f(V_{GS}, V_{DS_{sat}})$, see *figure 26*.

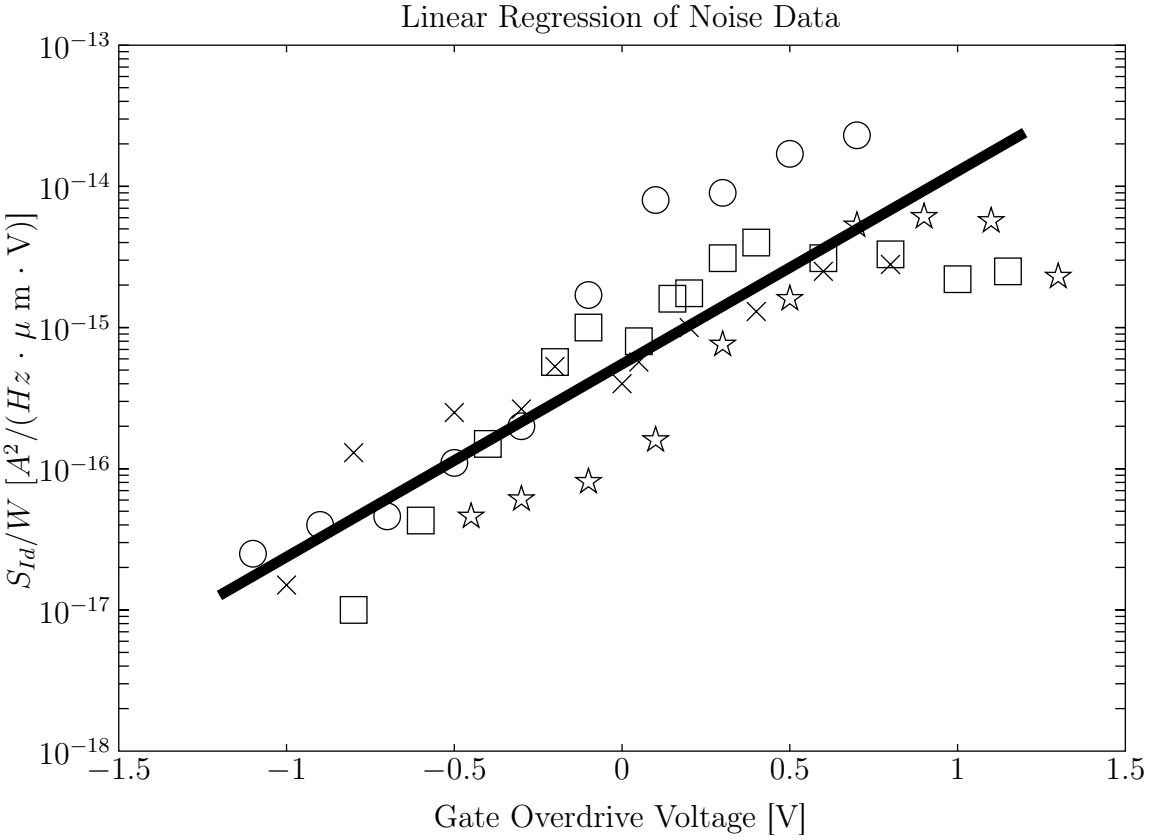

Figure 26: Linear regression of measured noise data for the NWFET.

From this regression, the flicker noise voltage dependence can be expressed as

$$f(V_{GS}, V_{DS_{sat}}) = 10^{1.37 \cdot (V_{GS} - V_t) - 15.26} \cdot V_{DS_{sat}} \tag{27}$$

Using the two built-in noise functions previously described, the source was implemented as:

```
I(d,s)<+white_noise(4*gamma*‘P_K*$temperature*(Id(V(g,s),V(d,s))/V(d,s)));

I(d,s)<+flicker_noise(nNW*(diam*pi*1e-3)*pow(10,(1.37*(V(g,s)-Vt)-15.26))*VdsSat(V(g,s), V
    (d,s)), 1);
```

where *VdsSat* is a new function introduced to calculate the saturated drain-source voltage.

The new improved noise model is described in *figure 27*.

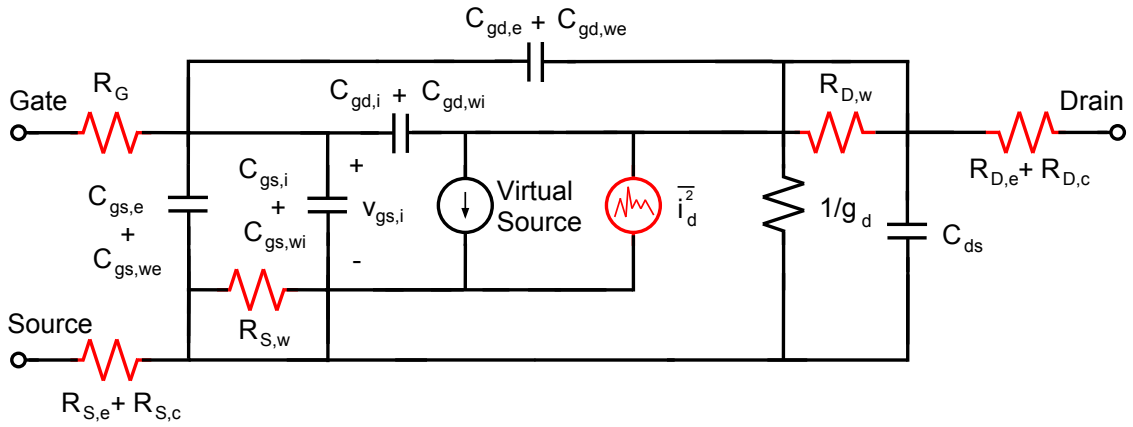

Figure 27: Source of noise in the new model.

3.2.1 Noise Contributions

In order to better understand the contribution from the different resistive elements and the implemented noise model of the channel (see *figure 28*), extensive noise summaries for the entire mixer for different values of γ were done. The results are presented in table 7. In the *RF FETs* and *LO FETs* columns, the noise is divided into channel thermal noise and noise from the small signal elements (the sum of noise from R_G, $R_{S,e}$, $R_{S,c}$, $R_{s,w}$, $R_{D,w}$, $R_{D,e}$ and $R_{D,c}$).

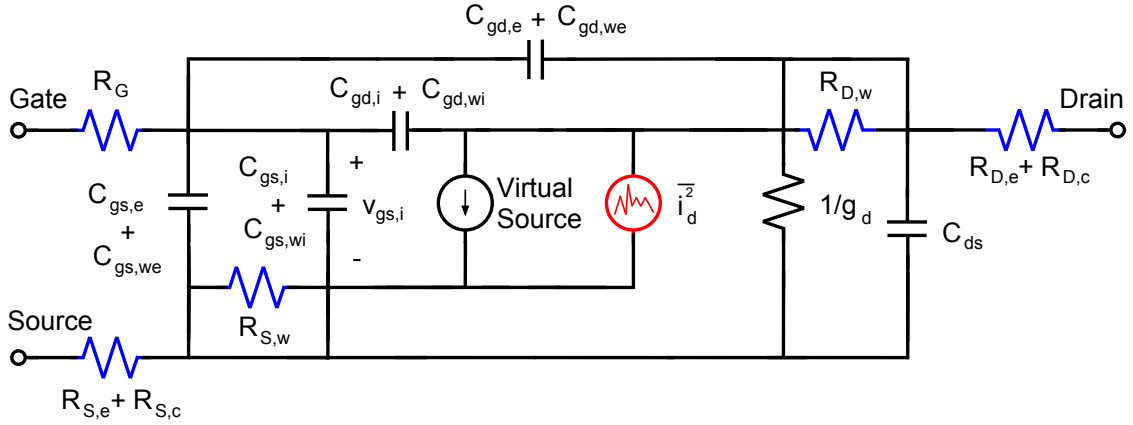

Figure 28: Distinction between contributions from the new noise source and the previous resistive elements.

3.2.2 Scaling Mixer to Reduce Noise

As mentioned in 2.6.3, the noise can be lowered by scaling the transistor sizes (number of wires), load resistors and bias current by a factor α.

With $\alpha = 1$ corresponding to mixer A (48 wires per transistor, 960 μA bias current and 1.6 $k\Omega$ load), the design was scaled for α $1 \rightarrow 20$ and NF_{DSB}, voltage conversion gain together with the DC power consumptions was simulated. γ was assumed to be 2.

3.3 Revised Transistor Layout for Larger Number of Wires

To investigate the effects behind the reduced mixer gain with an increased transistor scaling (*figure 34*) as series of DC sweeps where performed on three differently sized NWFETs (results shown in *figure 35*):

$\mathbf{n_{NW,y}}$	$\mathbf{n_{NW,x}}$	$\mathbf{n_{NW,x} \cdot n_{NW,y}}$	**Total Circumference [μm]**
8	6	48	7 μm
8	36	288	41 μm
8	120	960	136 μm

Table 3: Differently sized NWFETs with constant $n_{NW,y} = 8$.

In the case where $n_{NW,y}$ is fixed and the scaling is done by increasing $n_{NW,x}$, the ideal case is a symmetrical layout, hence values of $n_{NW,x}$ larger than $n_{NW,y}$ are discouraged:

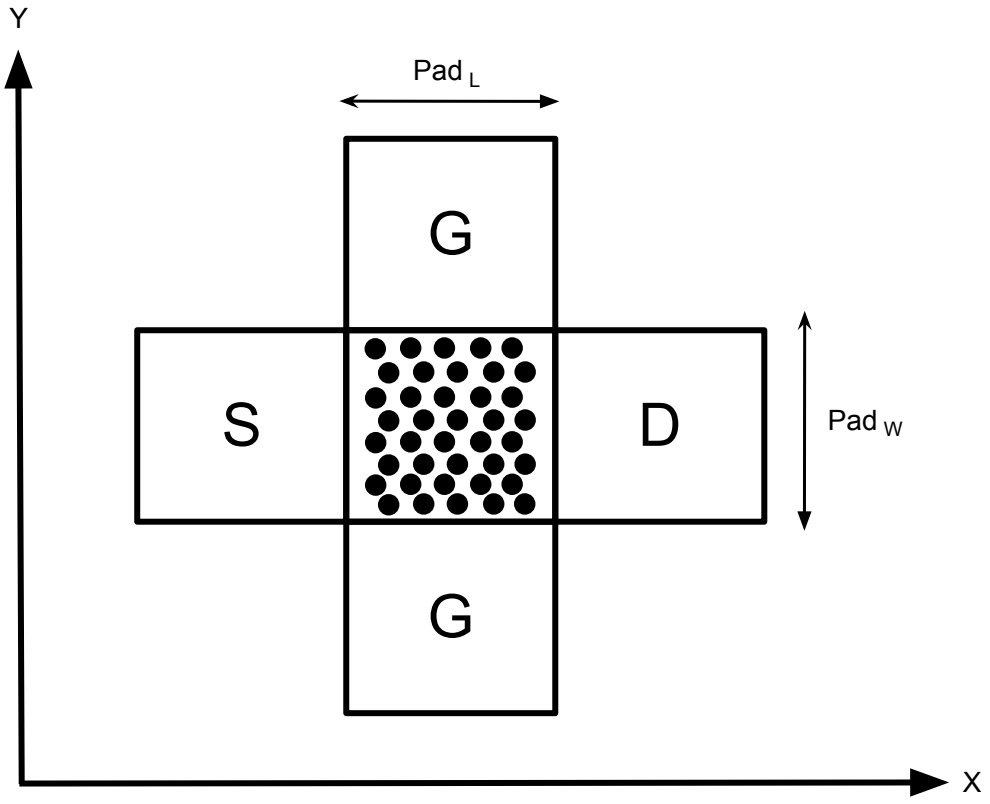

Figure 29: Symmetrical transistor layout with $n_{NW,x}$ close to $n_{NW,y}$.

The resistive elements in the NWFET small-signal model (*figure 2*) are dependent on the total number of sheets making up the contact pads. These dependencies are summarised in *table 4*.

Resistance	**Dependence on** $\frac{\mathbf{Pad_L}}{\mathbf{Pad_W}}$
$\mathbf{R_G}$	$5 \cdot 10^{-8} \cdot \frac{0.5 \cdot Pad_L}{0.06 \cdot 10^{-6} \cdot Pad_W}$
$\mathbf{R_{S,e}} + \mathbf{R_{S,c}}$	$R_\square \cdot 0.5 \cdot \left(1 + \frac{Pad_L}{Pad_W}\right)$
$\mathbf{R_{S,w}}$	*none*
$\mathbf{R_{D,e}} + \mathbf{R_{D,c}}$	*none*
$\mathbf{R_{D,w}}$	*none*

Table 4: Small-signal resistances and their layout dependence.

The scaling done in *3.2.2* kept $n_{NW,y}$ constant and increased $n_{NW,x}$. This will lead to a asymmetrical layout, causing the number of sheets ($\frac{Pad_L}{Pad_W}$) to increase compared to the symmetrical case:

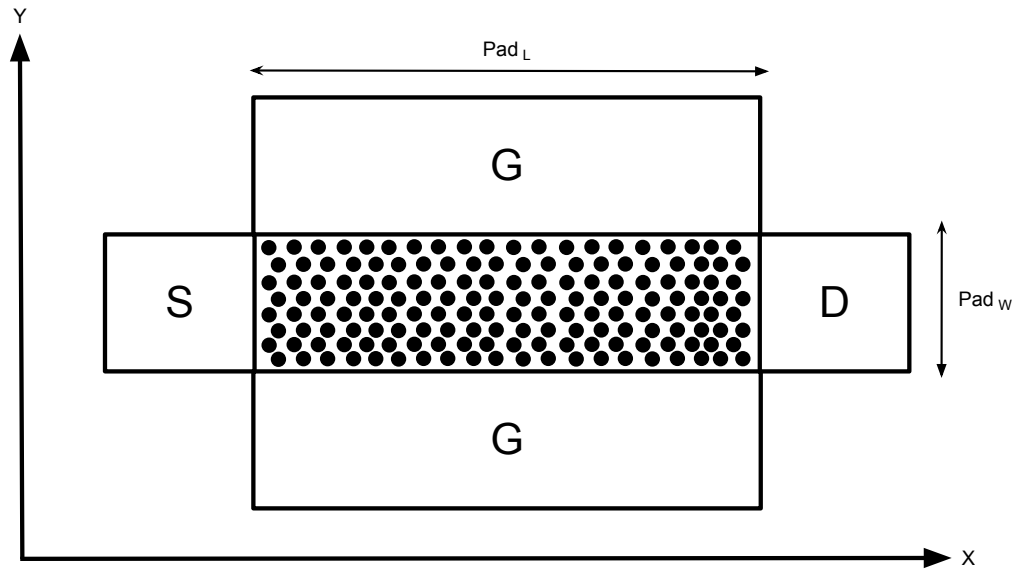

Figure 30: Asymmetrical transistor layout with $n_{NW,x} > n_{NW,y}$.

To investigate the differences between symmetrical and asymmetrical scaling, the same scaling as in *3.2.2* was performed again using both an unsymmetrical and symmetrical approach. The

resulting small-signal resistances were plotted. The results can be found in *figure 36* (asymmetrical case) and *figure 37* (symmetrical case).

The capacitances in the small-signal schematic are also dependent on the layout and the values used in the original model for this project was for $n_{NW,y} = 8$. To approximate the values for a scaled layout, the constant capacitances where scaled with a factor $n_{NW,y}/8$:

Capacitance	**Scaling ($\mathbf{n_{NW,y} = 8}$)**	**Suggested Scaling**
$\mathbf{C_{gs,e}}$	$C_{gs,e,C} + C_{gs,e,N} \cdot n_{NW,y}$	$C_{gs,e,C} \cdot n_{NW,y}/8 + C_{gs,e,N} \cdot n_{NW,y}$
$\mathbf{C_{gd,e}}$	$C_{gd,e,C} + C_{gd,e,N} \cdot n_{NW,y}$	$C_{gd,e,C} \cdot n_{NW,y}/8 + C_{gd,e,N} \cdot n_{NW,y}$
$\mathbf{C_{gs,i}}$	$C_{gs,i,N} \cdot n_{NW,y}$	*unchanged*
$\mathbf{C_{gd,i}}$	$C_{gd,i,N} \cdot n_{NW,y}$	*unchanged*
$\mathbf{C_{ds}}$	$C_{ds,e,C} + C_{ds,e,N} \cdot n_{NW,y}$	$C_{ds,e,C} \cdot n_{NW,y}/8 + C_{ds,e,N} \cdot n_{NW,y}$

Table 5: Small-signal capacitances and layout dependencies.

Using the scaling rules outlined here, a new transistor schematic and symbol was created in Cadence to incorporate these modifications. The same DC sweeps were performed on the new symmetric NWFET for approximately the same number of total wires as the previous sweep (49, 289, 961). The results are shown in *figure 38*. The transconductance as a function of gate voltage was simulated for a symmetrical and asymmetrical NWFET with $n_{NW} = 288$ and $V_{DS} = 0.3\ V$ (*figure 39*).

3.4 Design and Implementation of III-V NWFET-based Gilbert Cell

The goal is to first evaluate the two different performance enhancements against a mixer without enhancements for an RF range of 57 - 64 GHz and using the 50 nm NWFET node. The results from the enhanced designs will then be used to optimise a design combining both source degeneration and current bleeding, see *figure 31*.

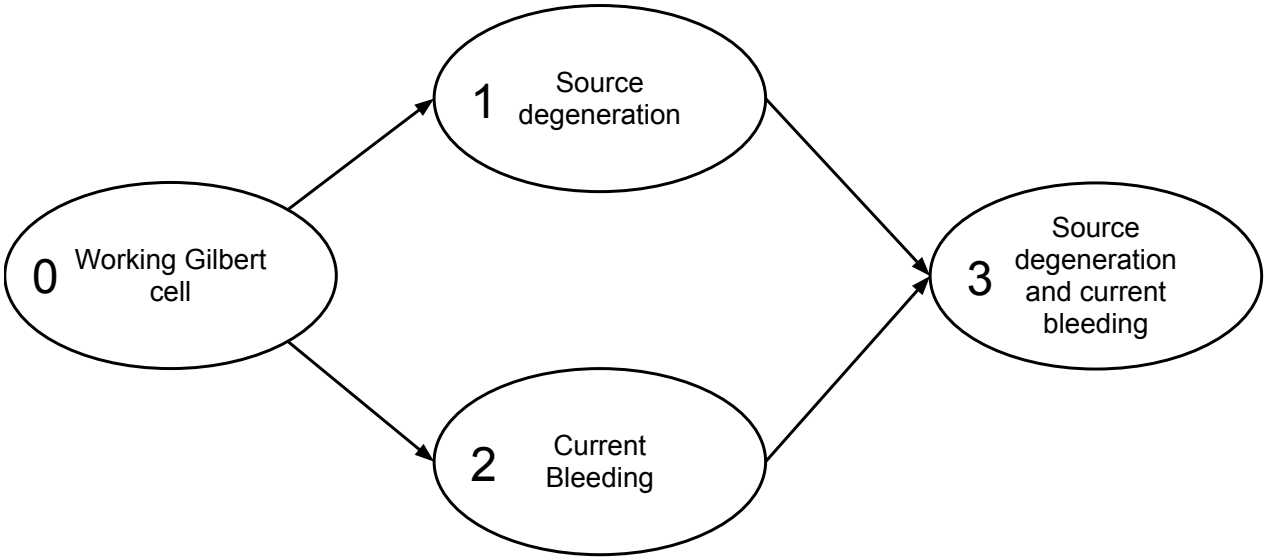

Figure 31: Proposed design flow.

The final design will then be evaluated once again, but this time for an RF range of 150 - 170 GHz. This will result in 5 unique mixers:

Mixer	RF Range [GHz]	NWFET Node [nm]	Enhancements
A	57 - 64	50	0
B	57 - 64	50	1
C	57 - 64	50	2
D	57 - 64	50	3
E	150 - 170	50	3

Table 6: Summary of the different mixers to design and benchmark.

3.4.1 Working Gilbert Cell (Mixer A)

Starting with an initial design corresponding to *figure 46* with the following parameters ($F_{IF} = 100\ MHz$):

$\mathbf{V_{DD}}$	$1.5\ V$
$\mathbf{R_L}$	$2.8\ k\Omega$
$\mathbf{n_{NW}}$	$8 \cdot 6 = 48$
$\mathbf{I_{tail}}$	$600\ \mu A$
$\mathbf{C_f}$	$\frac{1}{2\pi \cdot F_{IF} \cdot R_L}$

The schematic is outlined in *figure 46* in the appendix.

1. Using *Voltage Conversion Gain versus LO Signal Power* in the test bench, the voltage conversion gain was found to be 2.4 dB for P_{LO} of 8 dBm.
2. V_{DD} was kept at 1.5 V and R_L was swept $1\ k\Omega \rightarrow 5\ k\Omega$ and the bias current I_{tail} was swept $600\ \mu A \rightarrow 1.2\ mA$ in an attempt to optimise the conversion gain.
3. R_L was swept $1.6\ k\Omega \rightarrow 2.0\ k\Omega$ and the bias current I_{tail} was swept $900\ \mu A \rightarrow 1.0\ mA$. $R_L = 1.6\ k\Omega$ and $I_{tail} = 960\ \mu A$ yielded an improved conversion gain for the same P_{LO}.

The following new design parameters were chosen:

$\mathbf{V_{DD}}$	$1.5\ V$
$\mathbf{R_L}$	$1.6\ k\Omega$
$\mathbf{n_{NW}}$	$8 \cdot 6 = 48$
$\mathbf{I_{tail}}$	$960\ \mu A$
$\mathbf{C_f}$	$\frac{1}{2\pi \cdot F_{IF} \cdot R_L}$

This was the basis of the analysis in 3.2.2 . From the results of the noise analysis, the following final parameters were chosen ($\alpha = 6$). The previous value of the filter capacitance (C_f) was increased by a factor 3 to further increase conversion gain:

$\mathbf{V_{DD}}$	$1.5\ V$
$\mathbf{R_L}$	$270\ k\Omega$
$\mathbf{n_{NW}}$	$8 \cdot 36 = 288$
$\mathbf{I_{tail}}$	$5.8\ mA$
$\mathbf{C_f}$	$\frac{1}{2\pi \cdot 3 \cdot F_{IF} \cdot R_L}$

Finally, the full set of benchmarks in the automated test bench were performed.

3.4.2 Source Degeneration (Mixer B)

Starting with the final design of mixer A, with symmetrically scaled transistors and the following values:

$\mathbf{V_{DD}}$	$1.5\ V$
$\mathbf{R_L}$	$270\ \Omega$
$\mathbf{n_{NW}}$	$17 \cdot 17 = 289$
$\mathbf{I_{tail}}$	$5.8\ mA$
$\mathbf{C_f}$	$\frac{1}{2\pi \cdot 3 \cdot F_{IF} \cdot R_L}$
$\mathbf{L_{deg}}$	$0.5\ nH$

Where the values for the degenerative inductors (L_{deg}) was chosen based on the criteria

$$R_S \gg 1/g_m$$

which should correspond to the impedance

$$|Z| = 2\pi f_{RF} L_{deg} \gg 1/g_m$$

The schematic is outlined in *figure 47* in the appendix.

1. The value of the degenerative inductor was swept $0.5\ nH \rightarrow 2.0\ nH$ to investigate the impact on *IIP3* and voltage conversion gain

The results are shown in *figure 40*.

3.4.3 Current Bleeding (Mixer C)

Starting with the final design of mixer A, with symmetrically scaled transistors and the following values:

$\mathbf{V_{DD}}$	$1.5\ V$
$\mathbf{R_L}$	$270\ \Omega$
$\mathbf{n_{NW}}$	$17 \cdot 17 = 289$
$\mathbf{I_{tail}}$	$5.8\ mA$
$\mathbf{C_f}$	$\frac{1}{2\pi \cdot 3 \cdot F_{IF} \cdot R_L}$
$\mathbf{I_{bleed}}$	$I_{tail} \cdot C_{bleed}$

The schematic is outlined in *figure 48* in the appendix.

The constant C_{bleed} determines how much of the biasing current that will be diverted. Values as high as 50 % of the total current has been suggested in previous work [69].

1. C_{bleed} was swept $0 \rightarrow 0.25$ and voltage conversion gain, *NF*, *IIP3* and 1 dB compression point were simulated.

3.4.4 Source Degeneration and Current Bleeding (Mixer D & E)

These designs were omitted, due to the lack of conclusive results from the current bleeding design. See *5.4.4* for further discussion.

3.5 LNA Design Requirements

From (6), the overall noise ratio for a radio system is limited by the merits of its LNA and mixer stages. If the LNA and mixer can accomplish an overall conversion gain of 20-30 dB, then $A_{p_{LNA}} \cdot A_{p_{Mixer}} \approx 32$ and if the noise ratio for the rest of the system (NR_{Rest}) is sufficiently small in comparison, a rough estimation of the overall noise ratio ($NR_{Receiver}$) becomes

$$NR_{Receiver} \approx NR_{LNA} + \frac{NR_{Mixer} - 1}{A_{p_{LNA}}} \tag{28}$$

For a given mixer design it is thus possible to compute the requirements for a suitable LNA to keep the overall noise figure of the receiver ($NF_{Receiver}$) in an acceptable range (8-13 dB) and achieve an overall gain in the range 25-30 dB.

An implementation of this was done in MatLab (*LNA_Design.m*, 6). For a mixer with an NF of 14.6 dB and a voltage conversion gain of 3.47 dB:

```
LNA_design(14.6, 3.47);
```

The output for mixer A is shown in *figure 41* and for the case of sub-milliwatt power consumption, the result is shown in *figure 42*.

4 Results

4.1 Noise Model Implementation

4.1.1 Noise Contributions

γ	$\mathbf{NF_{DSB}}$ $[dB]$	Load	RF FETs	LO FETs
0	16.7	80 %	0 % + 11 %	0 % + 3 %
1	20.9	33 %	38 % + 5 %	20 % + 3 %
2	22.7	21 %	48 % + 3 %	25 % + 2 %
3	24.1	15 %	53 % + 2 %	28 % + 1 %
4	25.1	12 %	56 % + 2 %	29 % + 1 %
5	26.0	10 %	57 % + 1 %	30 % + 1 %

Table 7: $P_{LO} = 7\ dBm$ and $IF = 100\ MHz$.

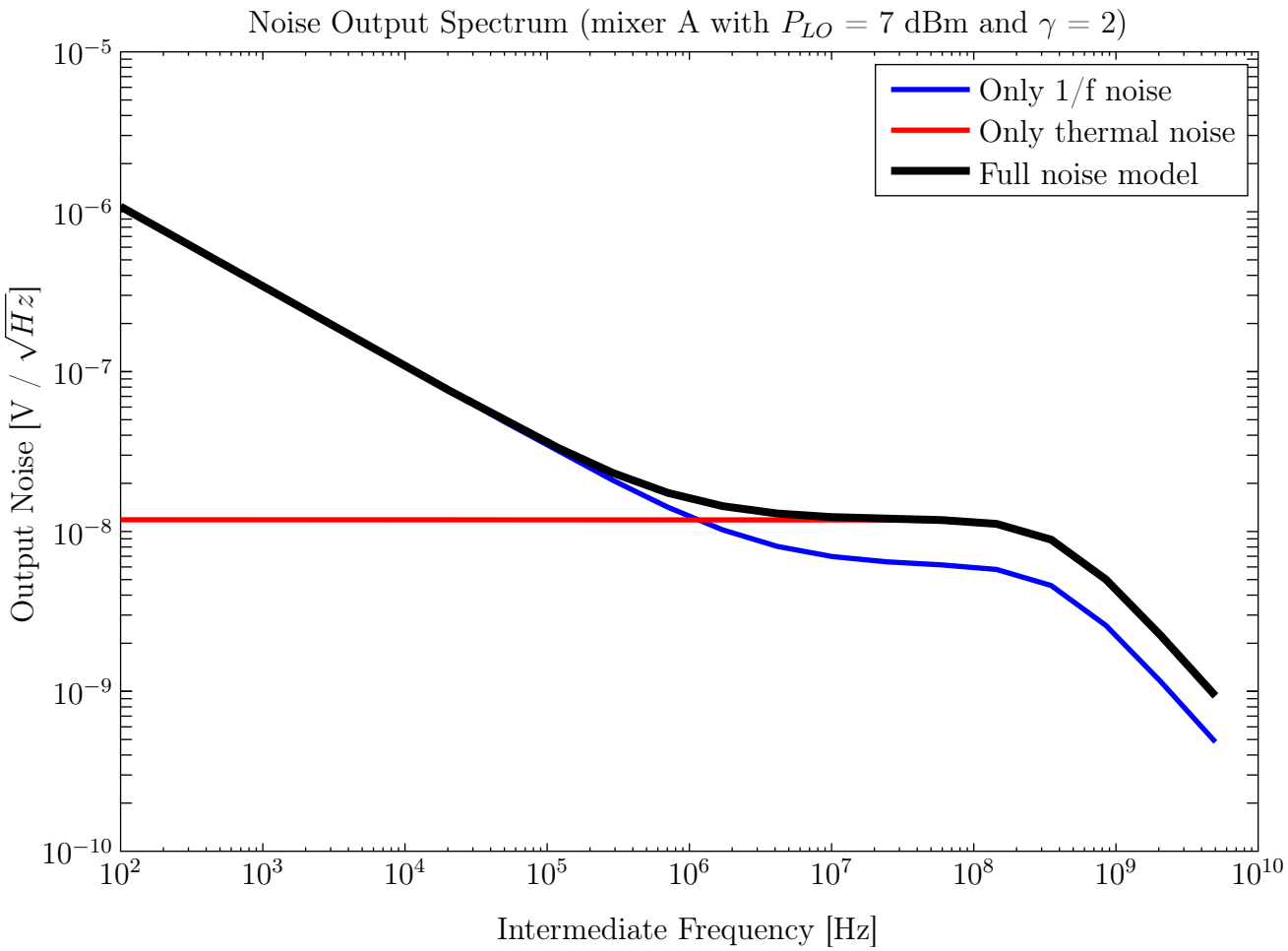

Figure 32: Noise output voltage spectrum.

4.1.2 Scaling Mixer to Reduce Noise

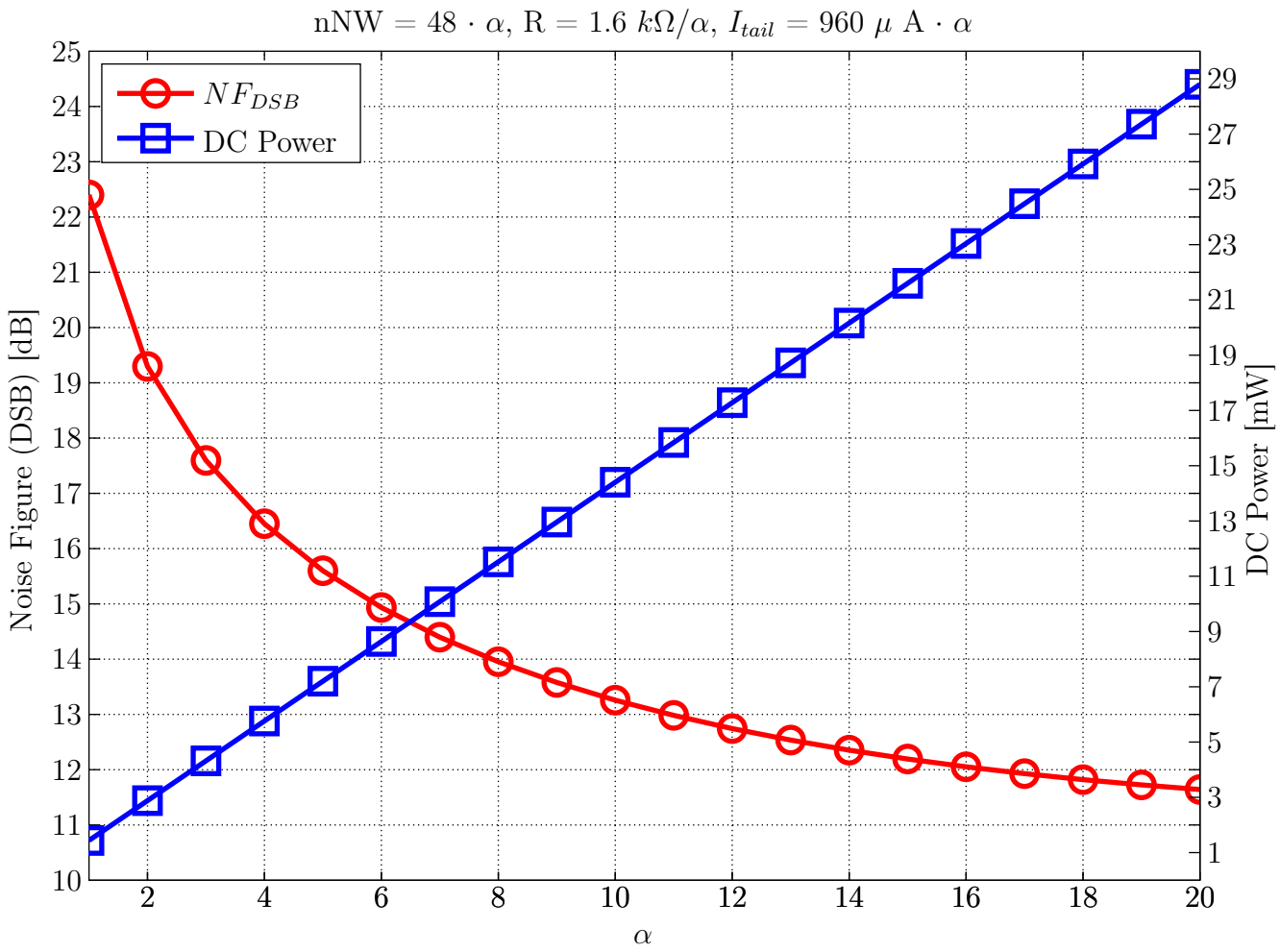

Figure 33: Device scaling effects on noise and DC power.

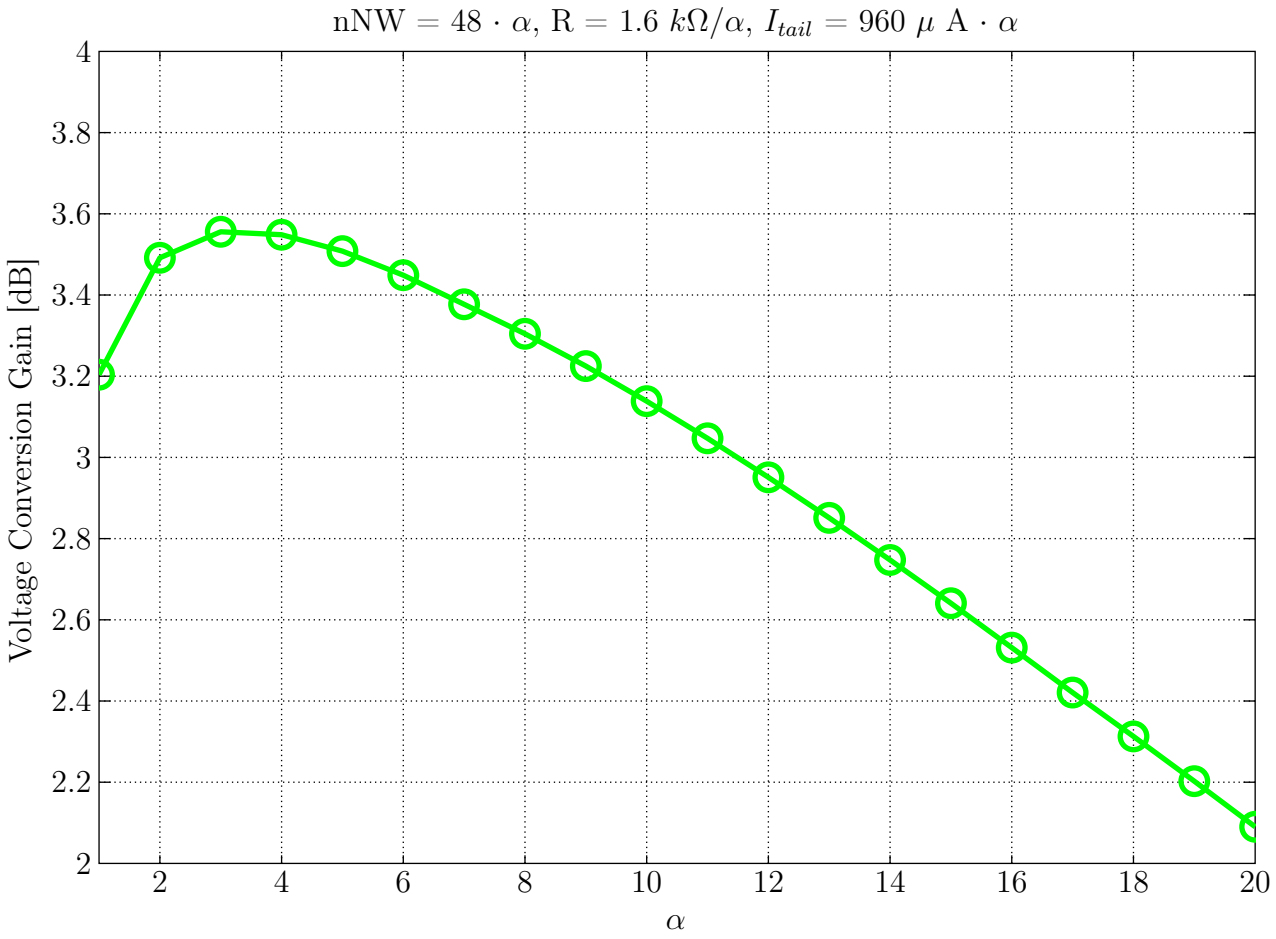

Figure 34: Device scaling effects on voltage conversion gain.

4.2 Revised Transistor Layout for Larger Number of Wires

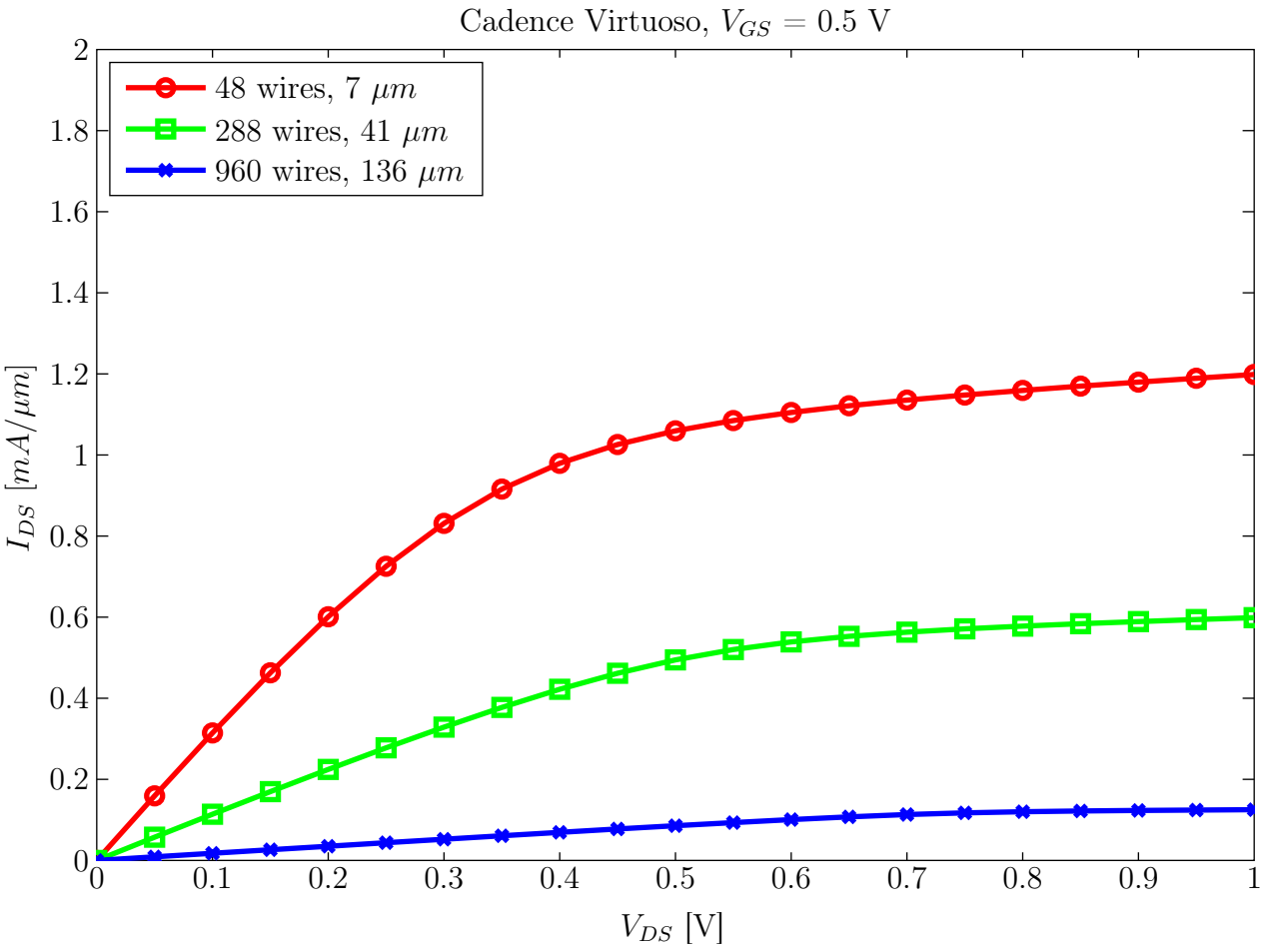

Figure 35: I_{DS} normalised to total wire circumference plotted against V_{DS}.

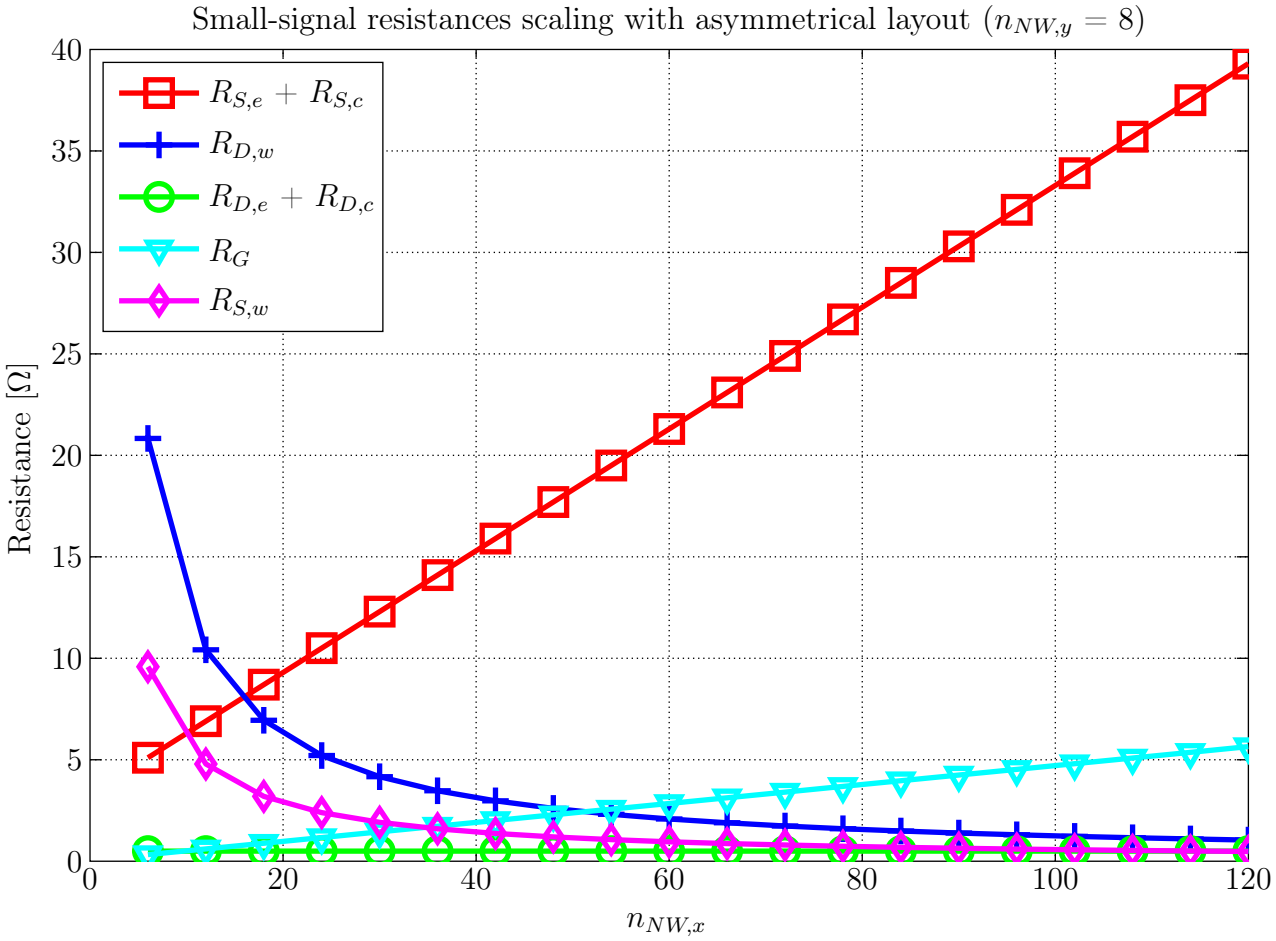

Figure 36: Small-signal resistor values for an asymmetrical scaling case.

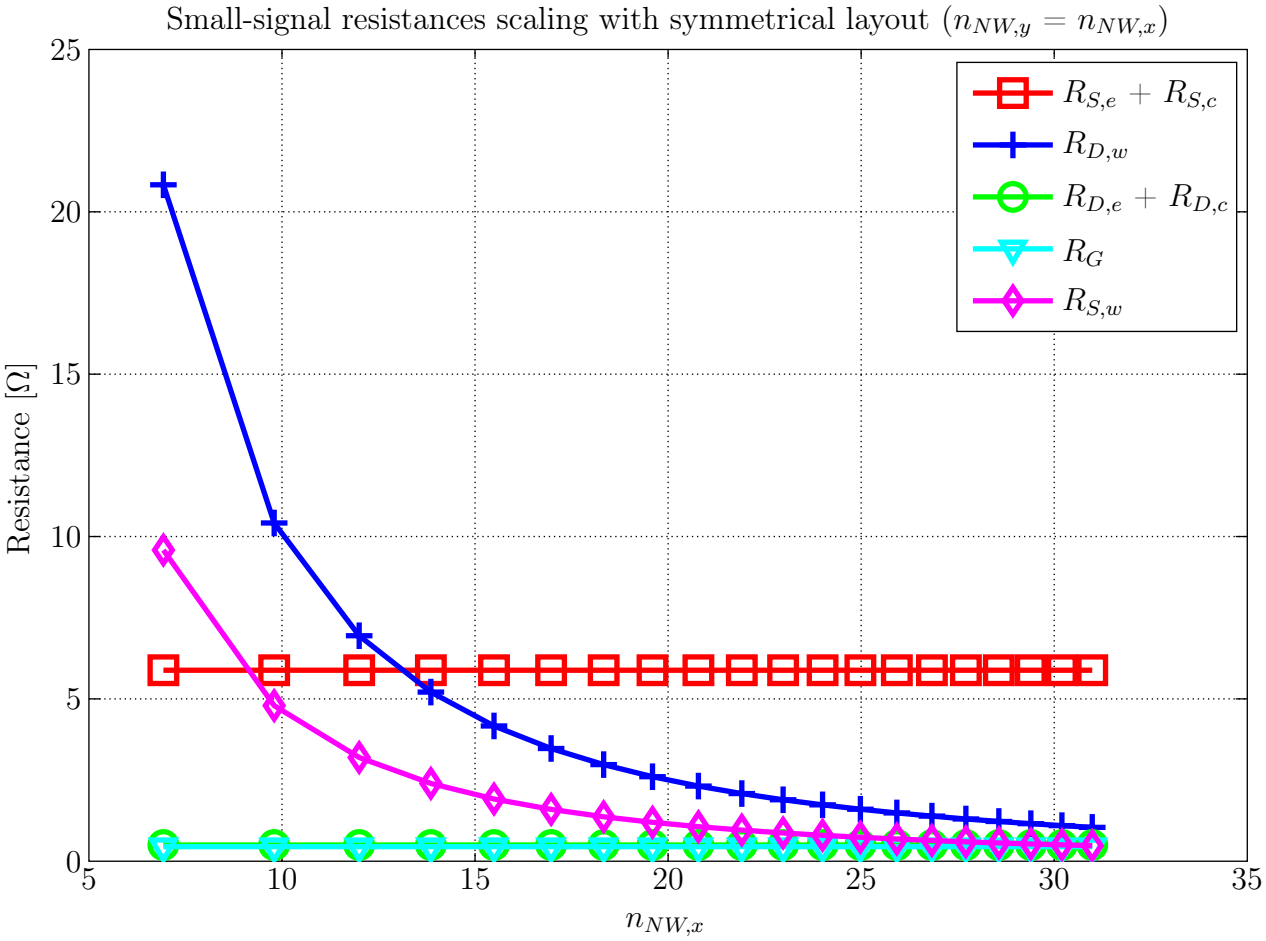

Figure 37: Small-signal resistor values for a symmetrical scaling case.

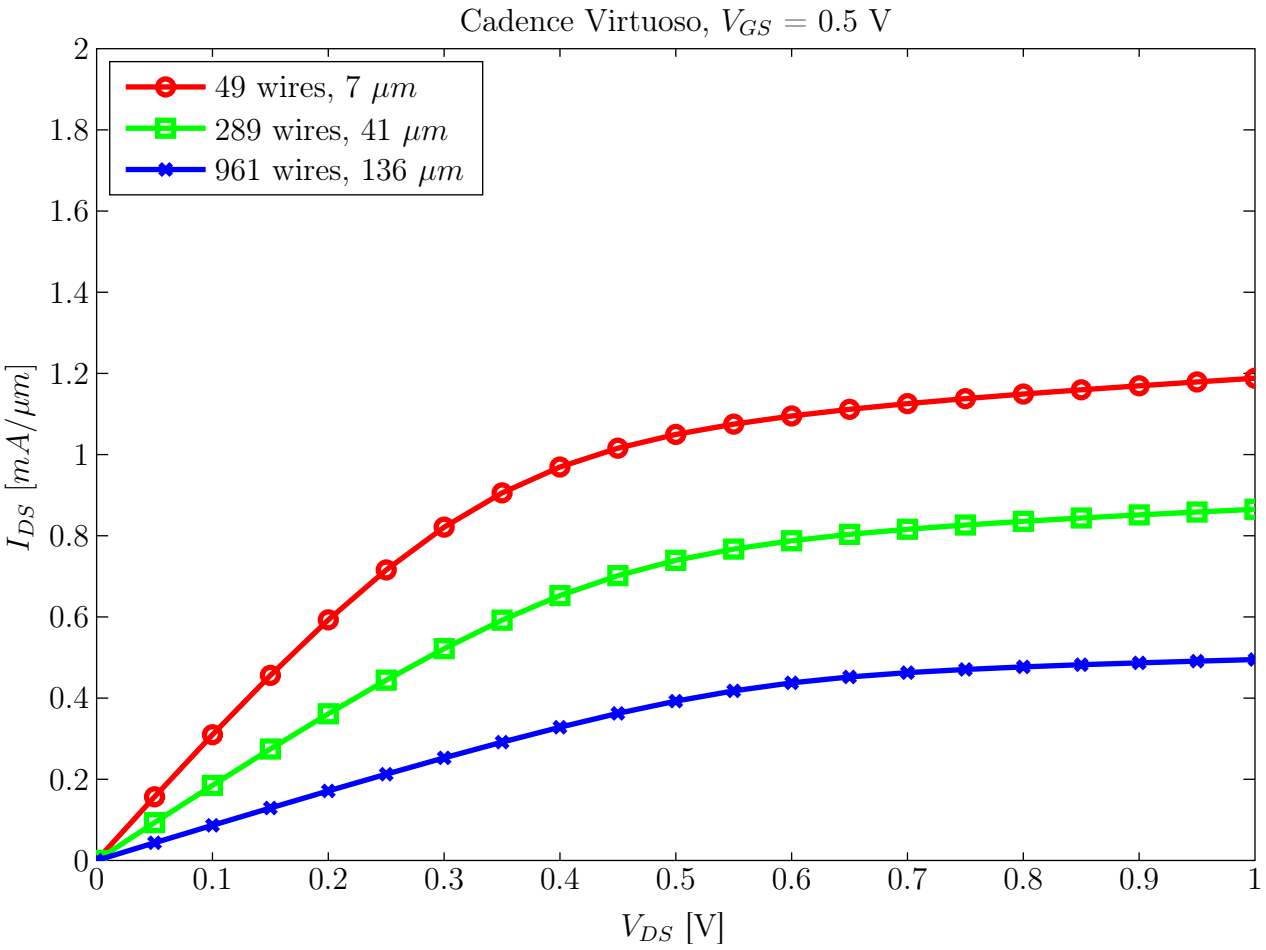

Figure 38: I_{DS} normalised to total wire circumference plotted against V_{DS} (improved transistor layout).

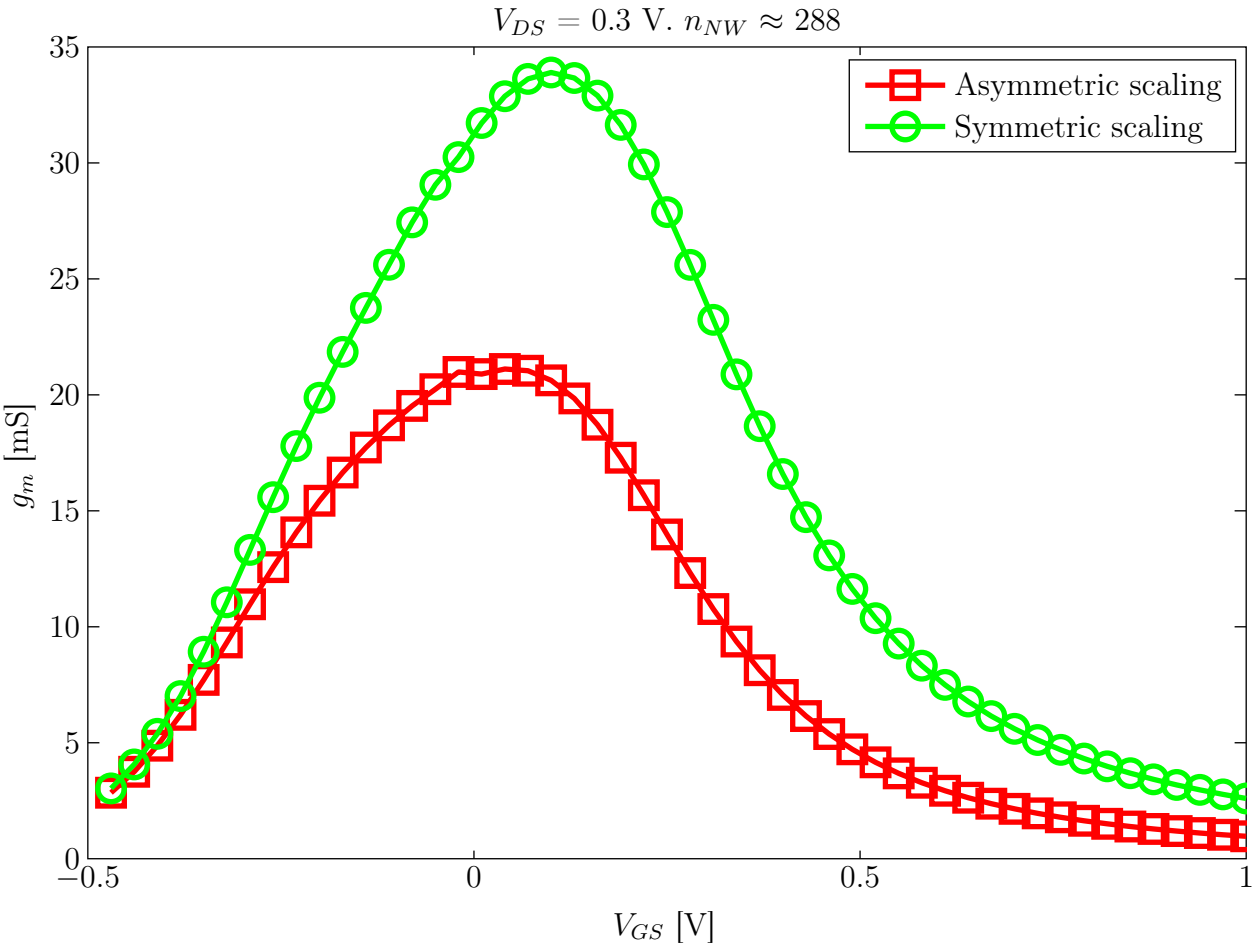

Figure 39: g_m plotted against V_{GS} for $V_{DS} = 0.3\ V$ for both symmetrical and asymmetrical transistor.

4.3 Mixer Design

4.3.1 Working Gilbert Cell (Mixer A)

RF Range [GHz]	57-64
Conversion Gain [dB]	3.47
NF @ 100 MHz IF [dB]	14.6
P_{1dB} [dBm]	0.4
IIP_3 [dBm]	8.5
DC Power [mW]	8.7
LO-RF Isolation [dB]	-52
LO-IF Isolation [dB]	-68

4.3.2 Source Degeneration (Mixer B)

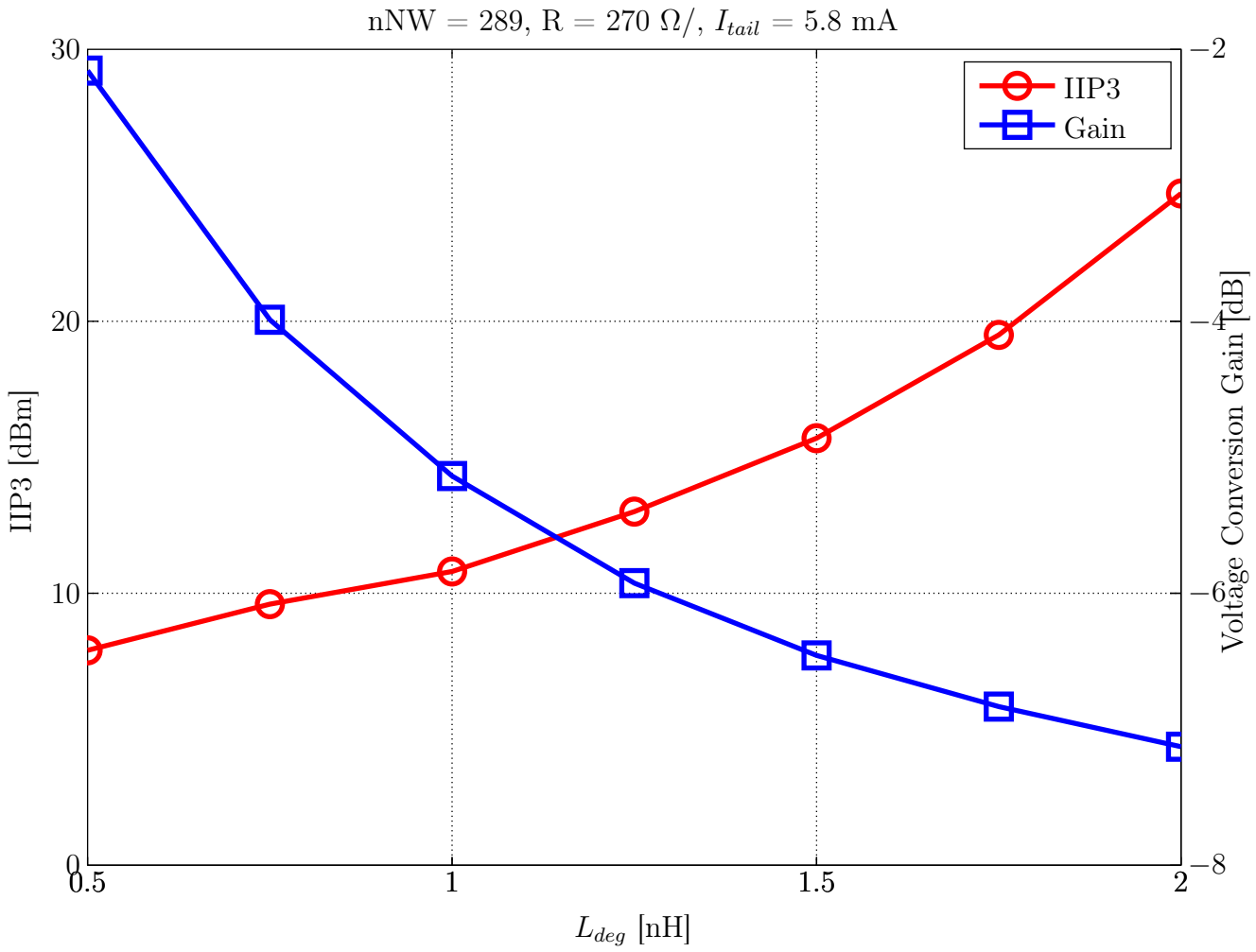

Figure 40: L_{deg} plotted against $IIP3$ and voltage conversion gain.

4.3.3 Current Bleeding (Mixer C)

The results for the current bleeding were inconclusive. See the discussion in *5.4.3*.

4.3.4 Source Degeneration and Current Bleeding (Mixer D & E)

As mentioned in the method outline, these designs were omitted, due to the lack of conclusive results from the current bleeding design. See *5.4.4* for further discussion.

4.4 LNA Design Requirements

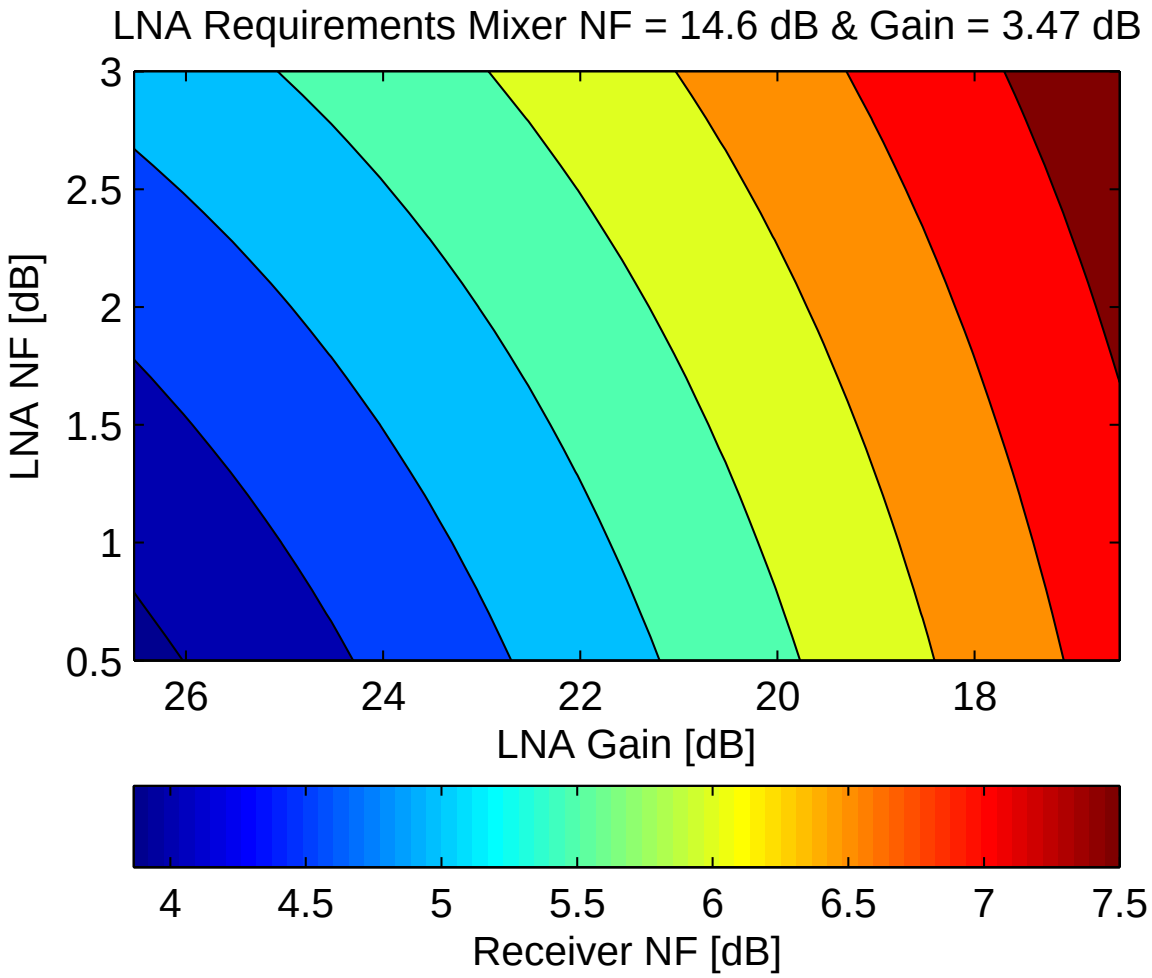

Figure 41: LNA design requirements for mixer A.

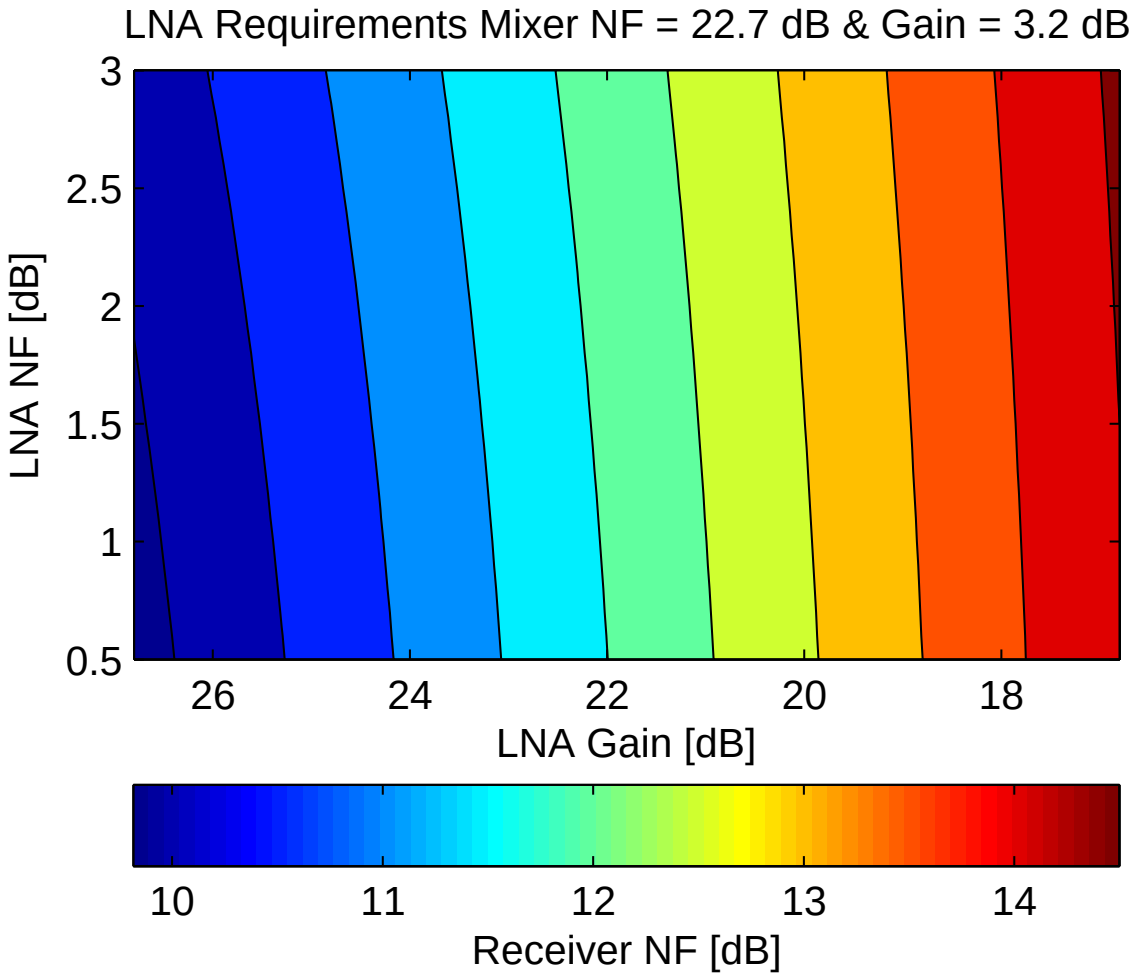

Figure 42: LNA design requirements for mixer A with sub-milliwatt power consumption.

5 Discussion

5.1 Automated RF Mixer Test Bench in Cadence Virtuoso

The test bench and the related OCEAN scripts form an versatile toolbox for future RF design work. For example, it is possible to re-use analysis concerning linearity, conversion gain and noise figure in future design efforts for a LNA.

It is also possible to simulate an entire radio front-end (LNA in combination with a mixer and a local oscillator) to evaluate the overall figures of merits.

Since the benchmarking has been standardised and highly repeatable, the simulations become an even more valuable tools when considering all the abstraction layers making up the final mixer circuits (*figure 43*).

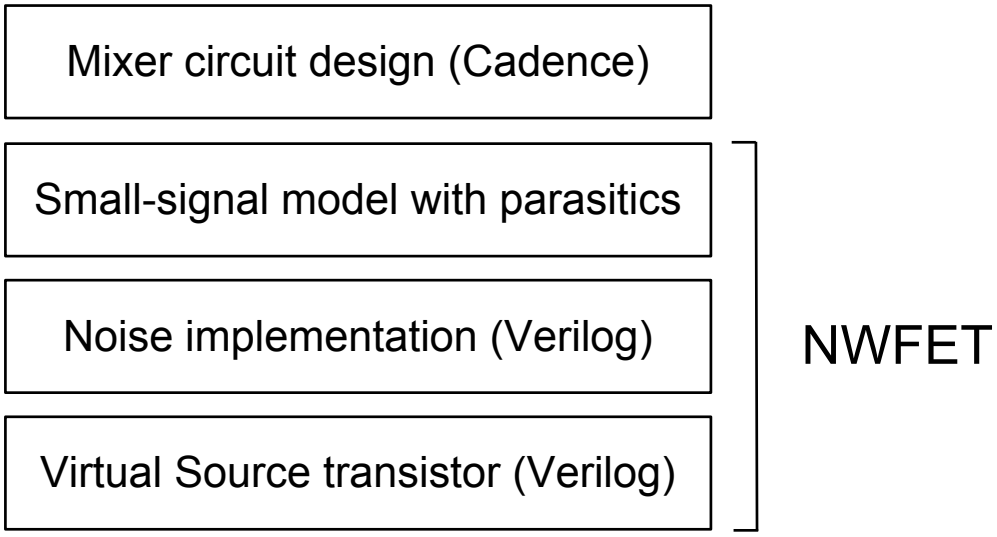

Figure 43: Abstraction layers in the overall design.

As the understanding of the physics and electrical properties of the NWFET increases, these independent models can be update accordingly and the entire high-level circuit evaluation can be repeated with ease.

5.1.1 Future Improvements

All the test cases, except *1 dB Compression and Third-Order Intercept*, rely on the RF signal source being configured as a DC source. In order to test linearity, a two-tone test is done, which requires the RF signal source to be switched to generate sine wave. This re-configuration is done in the procedure *uSetupRF*, but it has been noted that the changes might not always get picked up by Cadence Virtuoso.

Similarly, it is important to clear out the simulations folder and re-run the net-listing as big changes are made to the test bench, or when a new mixer is connected. Cadence Virtuoso might otherwise re-use simulations results, which can lead to strange results in the simulated figures of merits.

5.2 Implementation of Noise Model for the NWFET

Previous work by Karl-Magnus Persson at the department of electrical engineering, Lund University, Sweden on single-balanced mixers using the same NWFET compact model report a conversion gain of 6 db and a NF_{SSB} of 14 to 16 dB depending on the biasing conditions.

Before introducing a drain-source noise source, the computation of an overall NF for the mixer would only take into account thermal noise from resistive elements such as the load (R_L) and resistors in the small signal model. Simulations show that previously excluded thermal noise from the transistors contribute more to the overall noise than the aforementioned resistive components. The double balanced mixer is also expected to exhibit at least 3 dB higher noise figure than its single-balanced counterpart [54, p. 379]. Therefore, a simulated NF_{SSB} in the range of 20-25 dB might be realistic.

The degradation of NF due to the flicker noise of the mixing stage is a known limitation of MOSFET based current-commutating mixers at low baseband frequencies [70, 71]. With a frequency corner around 1 MHz, the flicker noise contribution is relatively small for IF frequencies above 100 MHz.

The noise voltage spectrum (*figure 32*) shows that the different components of the noise model are working as expected.

5.2.1 Verification by Calculations

Assuming a sine wave LO and $R_L = 2.8\ k\Omega$

$$A_v = \frac{1}{2} g_{m_{rf}} R_L = 2 \Rightarrow g_{m_{rf}} = \frac{4}{R_L} = 1.4\ mS \tag{29}$$

The noise figure due to thermal noise can be calculated from the input-referred noise voltage [54, p. 390]

$$\overline{V_{n,in}^2} = \pi^2 kT \left(\frac{\gamma}{g_{m_{rf}}} + \frac{2}{g_{m_{rf}}^2 R_L} \right) = 7.33 \cdot 10^{-17}\ V^2/Hz \tag{30}$$

with $\gamma = 2$ and $T = 300\ K$. The NF_{SSB} with respect to a R_S of 50 Ω is then

$$NF_{SSB} = 1 + \frac{\overline{V_{n,in}^2}}{4kTR_s} = 88.5 = 19.5\ dB \tag{31}$$

Thus, a NF_{SSB} of at least $19.5\ dB + 3\ dB = 22.5\ dB$ for a double-balanced mixer using equivalent biasing conditions and device sizes is to be expected. This value differs from the simulated NF_{SSB} of $25.7\ dB$, which can be explained by simple estimations of mixer noise typically underestimating the noise figure by $2 - 4\ dB$ [55, p. 22].

5.2.2 Accuracy

The flicker noise component of the noise has been implemented using experimental data from low-frequency noise measurements on single-wire NWFETs. For the overall noise of an array, it is assumed each wire in the array is fully functional and exhibit the exact same noise properties as a stand-alone wire. The total noise is thus the sum of all the noise contributions of the NWFETs in the array.

The thermal noise of the transistors is simulated for a γ of 2, which is the common approach for silicon MOSFET short-channel devices. Whether γ should be adjusted or not can be determined by future experiments carried out to better understand the thermal noise of the NWFETs.

It is also possible that the traditional MOSFET noise equation (23) alone is not enough to express the total noise. For example, shot noise due to gate leakage is omitted.

In Verilog it is not possible to get the derivative using built-in functions, although such functionality has been proposed [72]. To get the true value of g_m or g_d for noise calculations or other purposes the derivative would have to be expressed in existing terms corresponding to an analytical expression of the derivative. Because of these limitations, the $\frac{I_{DS}}{V_{DS}}$ term used in the presented noise model is an approximate expression of $\frac{\delta I_{DS}}{\delta V_{DS}}$.

The error in this approximation will grow larger as the transistor enters deep into saturation. For devices biased around the onset of saturation, which is the case for the mixer circuits used in this project, the error is thought to be small enough for practical purposes.

5.3 Revised Transistor Layout for Larger Number of Wires

The investigation of transistor scaling was done in order to understand the behaviour behind *figure 34*, where the voltage conversion gain is expected to increase as the mixer is scaled.

From *figure 35*, it becomes apparent that the biasing conditions requirements change as the transistors are made larger. In order to keep the transistors in saturation, as is the requirement for the gilbert mixer, the biasing would have to be re-done. This could explain the loss of voltage conversion gain seen in *figure 34*.

Another interesting observation is that depending on whether the transistor is made larger through asymmetric or symmetric scaling, the small-signal resistances will increase in the asymmetric case (*figure 36*) and remain constant in the symmetric case (*figure 37*).

Comparing *figure 35* to *figure 38*, the symmetric scaling makes the difference between required biasing conditions smaller, but still large enough to require new biasing as the transistors are scaled. Thus, the fall off in voltage conversion gain in *figure 34* is not solely due to the asymmetric transistor scaling.

Figure 39 shows how, for a relatively large transistor ($n_{NW} \approx 288$), the increased small-signal resistances due to the asymmetric scaling have a rather big influence on the overall transistor performance.

5.3.1 Maximum Achievable Voltage Conversion Gain

It is also worth pointing out that the maximum achievable voltage conversion gain is not as simple as expressed in (16), but actually [54, p. 376]

$$A_{V,max} = \frac{2}{\pi} g_{m1} R_L \left[1 - \frac{2V_{ov}}{5\pi V_{P,LO}}\right] \frac{g_{m2}}{\sqrt{C_p^2 \omega^2 + g_{m2}^2}} \tag{32}$$

where the capacitance C_p is

$$C_p = C_{DB1} + C_{GS2} + C_{GS3} + C_{SB2} + C_{SB3} \tag{33}$$

making up the capacitance seen from the point P (*figure 44*).

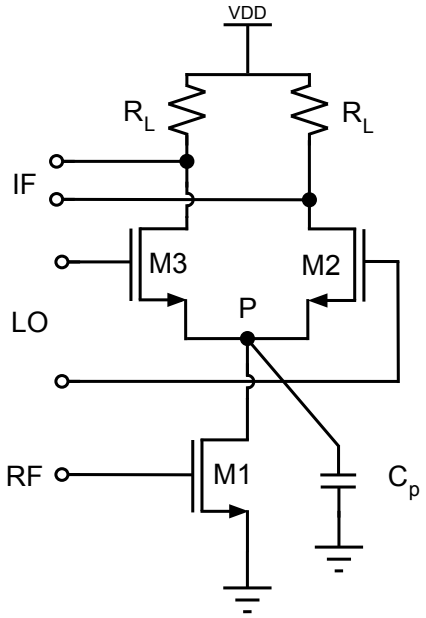

Figure 44: Limiting capacitance exemplified with a single-balanced mixer.

5.4 Design and Implementation of III-V NWFET-based Gilbert Cell

The design techniques evaluated in this thesis were chosen before the initial design of the mixer. Ideally, an early first mixer design would have made it possible to better select design techniques for improving the weak points in the III-V NWFET mixer design, such as noise performance.

From the literature studies, it became apparent that most of the design improvements used for silicon CMOS mixers are used to improve linearity. It seems that for these frequencies close to half the cut-off frequency, linearity might be a big issue for the silicon CMOS based RF components (as can be seen in the summary of 60 GHz receiver systems in *table 9*).

The linearity of the initial mixer design in this project turned out to be very good compared to its silicon CMOS counterparts, possibly on account of the higher cut-off frequency.

5.4.1 Working Gilbert Cell (Mixer A)

The figures of merits for mixer A were compared to similar mixers found in the literature:

	Mixer A	**[20]**	**[22]**
RF Range [GHz]	57-64	60	57-66
Conversion Gain [dB]	3.47	2	15.46
NF_{DSB} [dB]	14.6	15	12.8
	(100 MHz)	(600 MHz)	(100 MHz)
P_{1dB} [dBm]	0.4	-	-25
IIP_3 [dBm]	8.5	-6	-12
DC Power [mW]	8.7	6	17
V_{DD} [V]	1.5	1.2	1.35
LO-RF Isolation [dB]	-52	-	-64.7
LO-IF Isolation [dB]	-68	-	-51.5
Technology	III-V NWFET	65 nm CMOS	90 nm CMOS

Table 8: Comparing the figures of merits of mixer A with previous work.

The double-balanced mixer simulated here shows higher linearity than its silicon CMOS counterparts.

5.4.2 Source Degeneration (Mixer B)

As expected, *figure 40* clearly shows a decrease in gain and an increase of *IIP3* as the inductance L_{deg} is increased. For an inductance of 0.5 nH it can be noted that the mixer is already exhibiting conversion loss rather than gain.

From the table in *5.4.1* it can be seen that *IIP3* in the non-degenerated case is already superior to silicon CMOS counterparts.

5.4.3 Current Bleeding (Mixer C)

The simulations of a mixer with current bleeding were inconclusive and could not be compared to the theory outlined in *2.7.3*.

From the literature studies it appears that for a mixer where an increase in linearity is desirable, current bleeding makes it possible to maintain conversion gain while increasing *IIP3* [69].

In *"Current-reuse bleeding mixer"* by Lee and Choi [69], the effects of current bleeding were demonstrated by simulating and comparing a single-balanced mixer with and without current bleeding. The results showed an increase in conversion gain and *IIP3*, as well as an improved noise figure.

5.4.4 Source Degeneration and Current Bleeding (Mixer D & E)

In retrospect, the approach to combine source degeneration with current bleeding would only be desirable if the *IIP3* of the mixer was poor to begin with.

5.5 LNA Design Requirements

From the two cases *42* (sub-milliWatt mixer) and *41* (8.7 mW), there is a clear trade-off between low power operation and noise performance. Comparing the achievable receiver noise figure ranges $4 - 7.5\ dB$ to $10 - 14.5\ dB$ to previously reported receivers in the literature:

	[15]	[73]	[74]	[75]	[76]
Frequency Range [GHz]	57.5∼64	60∼58	57∼63	49∼53	52∼66
Conversion Gain [dB]	28	25∼30	55	26∼31	5∼14.5
Noise Figure [dB]	12.5	10∼12	4.9	7∼8	9.2
P_{1dB} [dBm]	-22.5	-27	-	-25.5	-24.4
DC Power [mW]	9	44	8	80	174
Area [mm²]	0.12	0.765	0.34	0.15	0.9
Type	Homodyne	Heterodyne	Homodyne	Heterodyne	Heterodyne
CMOS Technology	130 nm	130 nm	65 nm	90 nm	65 nm

Table 9: Previously reported 60 GHz silicon CMOS receiver front-ends.

The feasibility of targeting a total DC power consumption lower than achieved with 65 nm silicon CMOS, while keeping all other figures of merits comparable or improved, can be further investigated as other RF components such as the LNA and voltage controlled oscillator (VCO) are realised. Although compared with [74], which achieves a total noise figure of 4.9 dB at a total DC power lower than that for the entire mixer, there is clearly room for improvements.

6 Conclusions

- The double-balanced mixer topology provides better port isolation and is less sensitive to noise in the local oscillator signal than its single-balanced counterpart. However, these benefits have to be weighted against higher noise figure, lower conversion gain and higher power consumption.

- The need for balanced radio frequency inputs to the double-balanced mixer adds additional design complexity to the low noise amplifier. Baluns can be used to convert an unbalanced signal from the LNA, but at the cost of insertion loss lowering the overall gain of the mixer stage.

- There is a major trade-off between low power operation and good noise performance. The noise performance of the sub-milliwatt III-V NWFET double-balanced mixer significantly deteriorates the overall noise figure of a receiver, even with a superb LNA.

- Flicker noise seems to not be an issue for intermediate frequencies above 1 MHz. Currently the flicker noise component of the added noise model uses fitted data from measurements and is therefore considered more accurate than the white noise, which remains uncertain as to which value of γ to use.

- Transistor sizing is not as trivial as in the case with $\frac{W}{L}$ used in silicon CMOS. There is a dramatic loss of performance if the transistor sizing is done by asymmetric scaling, but the symmetric scaling attempted in this project uses rough approximations of small-signal resistances and capacitances that might be unrealistic.

- There is a need to bridge the gap between the design paradigm of traditional planar silicon CMOS design rules and the rules governing circuit design with III-V NWFETs in order to allow experienced circuit designers to make III-V NWFET based components truly comparable to their equivalent state-of-the-art silicon CMOS.

- Ultimately, the performance of an RF component, such as the mixer, can only be truly judged in comparison to the requirements of a communication system with respect to a standard such as IEEE 802.15.3c. The noise performance together with the linearity determines the sensitivity of a receiver, which has to be put in context with an overall link budget.

- Projects involving both modifications to the underlying transistor model and design work on the circuit level quickly tends to take longer time than expected, due to the inevitable overhead of iterating back and forth between the two; results from simulations on the circuit level become obsolete when the transistor model changes and the circuit design should not be performed with an unoptimised transistor model.

  On the other hand, an unoptimised model can help predict performance of circuits manufactured in the near future, whereas optimising the model helps exploring the upper bounds of what the technology is capable off.

References

[1] P.K. Bondyopadhyay. "Moore's Law Governs the Silicon Revolution". In: *Proceedings of the IEEE* 86.1 (1998), pp. 78 –81. DOI: 10.1109/5.658761.

[2] D.J. Frank et al. "Device Scaling Limits of Si MOSFETs and Their Application Dependencies". In: *Proceedings of the IEEE* 89.3 (2001), pp. 259–288. DOI: 10.1109/5.915374.

[3] Bo Yu et al. "Scaling of Nanowire Transistors". In: *Electron Devices, IEEE Transactions on* 55.11 (2008), pp. 2846 –2858. DOI: 10.1109/TED.2008.2005163.

[4] C.P. Auth and J.D. Plummer. "Scaling theory for cylindrical, fully-depleted, surrounding-gate MOSFETs". In: *IEEE Electron Device Letters* 18.2 (1997), pp. 74–76. DOI: 10.1109/55.553049.

[5] Kristofer Jansson, Erik Lind, and Lars-Erik Wernersson. "Performance Evaluation of III-V Nanowire Transistors". In: *IEEE Transactions on Electron Devices* 59.9 (2012), pp. 2375–2382. DOI: 10.1109/TED.2012.2204757.

[6] Sofia Johansson et al. "RF Characterization of Vertical InAs Nanowire Wrap-Gate Transistors Integrated on Si Substrates". In: *IEEE Transactions on Microwave Theory and Techniques* 59.10 (2011), pp. 2733–2738. DOI: 10.1109/TMTT.2011.2163076.

[7] Erik Lind and Lars-Erik Wernersson. "Design of RF Properties for Vertical Nanowire MOSFETs". In: *IEEE Transactions on Nanotechnology* 10.4 (2011), pp. 668–673. DOI: 10.1109/TNANO.2010.2064783.

[8] A.D. Olver. "Millimetrewave Systems - Past, Present and Future". In: *Radar and Signal Processing, IEE Proceedings F* 136.1 (1989), pp. 35 –52.

[9] T. Baykas et al. "IEEE 802.15.3c: The First IEEE Wireless Standard for Data Rates Over 1 Gb/s". In: *Communications Magazine, IEEE* 49.7 (2011), pp. 114 –121. DOI: 10.1109/MCOM.2011.5936164.

[10] A. Hajimiri. "mm-Wave Silicon ICs: Challenges and Opportunities". In: *Custom Integrated Circuits Conference, 2007. CICC '07. IEEE*. 2007, pp. 741 –747. DOI: 10.1109/CICC.2007.4405837.

[11] Jim Martin et al. "WiMAX Performance at 4.9 GHz". In: *2010 IEEE Aerospace Conference*. Institute of Electrical and Electronics Engineers, 2010, pp. 1–8. DOI: 10.1109/AERO.2010.5446943.

[12] Robert Daniels and Robert Heath Jr. "60 GHz Wireless Communications: Emerging Requirements and Design Recommendations". In: *IEEE Vehicular Technology Magazine* 2.3 (2007), pp. 41–50. DOI: 10.1109/MVT.2008.915320.

[13] M. Kraemer et al. "A Wideband Single-balanced Down-mixer for the 60 GHz Band in 65 nm CMOS". In: *Microwave Conference Proceedings (APMC), 2010 Asia-Pacific*. 2010, pp. 1849 –1852.

[14] S. Emami et al. "A 60-GHz Down-converting CMOS Single-gate Mixer". In: *RFIC) Symposium, 2005. Digest of Papers. 2005 IEEE Radio Frequency integrated Circuits*. Institute of Electrical and Electronics Engineers, 2005, pp. 163–166. DOI: 10.1109/RFIC.2005.1489619.

[15] B. Razavi. "A 60-GHz CMOS Receiver Front-End". In: *IEEE Journal of Solid-State Circuits* 41.1 (2006), pp. 17–22. DOI: 10.1109/JSSC.2005.858626.

[16] Jeng-Han Tsai et al. "A 25-75 GHz Broadband Gilbert-Cell Mixer Using 90-nm CMOS Technology". In: *IEEE Microwave and Wireless Components Letters* 17.4 (2007), pp. 247–249. DOI: 10.1109/LMWC.2007.892934.

[17] Fan Zhang, Efstratios Skafidas, and William Shieh. "A 60-GHz Double-Balanced Gilbert Cell Down-Conversion Mixer on 130-nm CMOS". In: *2007 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*. Institute of Electrical and Electronics Engineers, 2007, pp. 141–144. DOI: 10.1109/RFIC.2007.380851.

[18] Mikko Kantanen et al. "60 GHz Frequency Conversion 90 nm CMOS Circuits". In: *2008 European Microwave Integrated Circuit Conference*. Institute of Electrical and Electronics Engineers, 2008, pp. 60–63. DOI: 10.1109/EMICC.2008.4772228.

[19] Dong-Hyun Kim and Jae-Sung Rieh. "A Single-balanced 60-GHz Down-conversion Mixer in 0.13 μ m CMOS Technology for WPAN Applications". In: *2009 34th International Conference on Infrared, Millimeter, and Terahertz Waves*. Institute of Electrical and Electronics Engineers, 2009, pp. 1–2. DOI: 10.1109/ICIMW.2009.5325647.

[20] P. Sakian et al. "A 60-GHz Double-balanced Homodyne Down-converter in 65-nm CMOS Process". In: *Microwave Integrated Circuits Conference, 2009. EuMIC 2009. European*. 2009, pp. 258 –261.

[21] M. Ercoli et al. "A Passive Mixer for 60 GHz Applications in CMOS 65nm Technology". In: *German Microwave Conference, 2010*. 2010, pp. 20 –23.

[22] J.H. Lee and Y.S. Lin. "60 GHz CMOS Downconversion Mixer with 15.46 dB gain and 64.7 dB LO-RF Isolation". In: *Electronics Letters* 49.4 (2013), pp. 264–266. DOI: 10.1049/el.2012.4165.

[23] Thomas Lee. *The design of CMOS radio-frequency integrated circuits*. Cambridge, UK New York: Cambridge University Press, 2004.

[24] T. Yao et al. "60-GHz PA and LNA in 90-nm RF-CMOS". In: *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium, 2006*. Institute of Electrical and Electronics Engineers, 2006, pp. 125–128. DOI: 10.1109/RFIC.2006.1651107.

[25] S.T. Nicolson et al. "Design and Modeling Considerations for Fully-Integrated Silicon W-Band Transceivers". In: *2007 IEEE International Workshop on Radio-Frequency Integration Technology*. Institute of Electrical and Electronics Engineers, 2007, pp. 141–149. DOI: 10.1109/RFIT.2007.4443938.

[26] J.-O. Plouchart et al. "A 243-GHz F_t and 208-GHz F_{max}, 90-nm SOI CMOS SoC Technology With Low-Power mmWave Digital and RF Circuit Capability". In: *IEEE Transactions on Electron Devices* 52.7 (2005), pp. 1370–1375. DOI: 10.1109/TED.2005.850638.

[27] Jean-Olivier Plouchart. "Applications of SOI Technologies to Communication". In: *2011 IEEE Compound Semiconductor Integrated Circuit Symposium (CSICS)*. Institute of Electrical and Electronics Engineers, 2011, pp. 1–4. DOI: 10.1109/CSICS.2011.6062437.

[28] H. Rucker et al. "A 0.13 μm SiGe BiCMOS Technology Featuring f_T/f_{max} of 240/330 GHz and Gate Delays Below 3 ps". In: *2009 IEEE Bipolar/BiCMOS Circuits and Technology Meeting*. Institute of Electrical and Electronics Engineers, 2009, pp. 166–169. DOI: 10.1109/BIPOL.2009.5314251.

[29] G. K. Celler and Sorin Cristoloveanu. "Frontiers of Silicon-on-insulator". In: *Journal of Applied Physics* 93.9 (2003), p. 4955. DOI: 10.1063/1.1558223.

[30] Ozgur Inac, Andy Fung, and Gabriel M. Rebeiz. "Double-balanced 130-180 GHz Passive and Balanced 145-165 GHz Active Mixers in 45 nm CMOS". In: *2011 IEEE Custom Integrated Circuits Conference (CICC)*. Institute of Electrical and Electronics Engineers, 2011, pp. 1–4. DOI: 10.1109/CICC.2011.6055323.

[31] Sungjae Lee et al. "Record RF Performance of 45-nm SOI CMOS Technology". In: *2007 IEEE International Electron Devices Meeting*. Institute of Electrical and Electronics Engineers, 2007, pp. 255–258. DOI: 10.1109/IEDM.2007.4418916.

[32] A.J. Joseph et al. "Status and Direction of Communication Technologies - SiGe BiCMOS and RFCMOS". In: *Proceedings of the IEEE* 93.9 (2005), pp. 1539–1558. DOI: 10.1109/JPROC.2005.852547.

[33] Dong-Hyun Kim and Jae-Sung Rieh. "A SiGe 140-GHz Low Power G_m-boosted Down-conversion Mixer". In: *Microwave Conference Proceedings (APMC), 2011 Asia-Pacific*. 2011, pp. 1126 –1129.

[34] Dong-Hyun Kim and Jae-Sung Rieh. "A 135 GHz Differential Active Star Mixer in SiGe BiCMOS Technology". In: *IEEE Microwave and Wireless Components Letters* 22.8 (2012), pp. 409–411. DOI: 10.1109/LMWC.2012.2205909.

[35] Ekaterina Laskin et al. "165-GHz Transceiver in SiGe Technology". In: *IEEE Journal of Solid-State Circuits* 43.5 (2008), pp. 1087–1100. DOI: 10.1109/JSSC.2008.920336.

[36] E. Laskin et al. "80/160-GHz Transceiver and 140-GHz Amplifier in SiGe Technology". In: *2007 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*. Institute of Electrical and Electronics Engineers, 2007, pp. 153–156. DOI: 10.1109/RFIC.2007.380854.

[37] Ullrich R Pfeiffer, Erik Ojefors, and Yan Zhao. "A SiGe Quadrature Transmitter and Receiver Chipset for Emerging High-frequency Applications at 160GHz". In: *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*. Institute of Electrical and Electronics Engineers, 2010, pp. 416–417. DOI: 10.1109/ISSCC.2010.5433832.

[38] Joseph J. Carr. "3 - Radio receivers". In: *RF Components and Circuits*. Oxford: Newnes, 2002, pp. 34 –65. DOI: 10.1016/B978-075064844-8/50004-9.

[39] A. Loke and F. Ali. "Direct conversion radio for digital mobile phones-design issues, status, and trends". In: *IEEE Transactions on Microwave Theory and Techniques* 50.11 (2002), pp. 2422–2435. DOI: 10.1109/TMTT.2002.804624.

[40] Cadence Design Systems, Inc. *Mixer Design Using SpectreRF Application Note*. Cadence Design Systems, Inc., 2004.

[41] Hsuan-Ling Kao et al. "A 30-65 GHz Wideband Double-balanced Gilbert-cell Mixer Using GaAs pHEMT Technology". In: *Microwave and Optical Technology Letters* 54.5 (2012), pp. 1196–1200. DOI: 10.1002/mop.26773.

[42] Dong-Hyun Kim and Jae-Sung Rieh. "CMOS 138 GHz Low-power Active Mixer with Branch-line Coupler". In: *Electronics Letters* 48.10 (2012), p. 554. DOI: 10.1049/el.2012.0402.

[43] Ozgur Inac et al. "Millimeter-Wave and THz Circuits in 45-nm SOI CMOS". In: *2011 IEEE Compound Semiconductor Integrated Circuit Symposium (CSICS)*. Institute of Electrical and Electronics Engineers, 2011, pp. 1–4. DOI: 10.1109/CSICS.2011.6062430.

[44] Xiongliang Lai and Fei Yuan. "A Comparative Study of Low-power CMOS Gilbert Mixers in Weak and Strong Inversion". In: *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*. Institute of Electrical and Electronics Engineers, 2011, pp. 1–4. DOI: 10.1109/MWSCAS.2011.6026681.

[45] Jinsung Park et al. "Design and Analysis of Low Flicker-Noise CMOS Mixers for Direct-Conversion Receivers". In: *IEEE Transactions on Microwave Theory and Techniques* 54.12 (2006), pp. 4372–4380. DOI: 10.1109/TMTT.2006.885582.

[46] G. Kathiresan and C. Toumazou. "A Low Voltage Bulk Driven Downconversion Mixer Core". In: *ISCAS99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349)*. Institute of Electrical and Electronics Engineers, 1999, pp. 598–601. DOI: 10.1109/ISCAS.1999.780831.

[47] Daryl Van Vorst and Shahriar Mirabbasi. "Low-power 1V 5.8 GHz Bulk-driven Mixer with on-chip Balun in 0.18μ m CMOS". In: *2008 IEEE Radio Frequency Integrated Circuits Symposium*. Institute of Electrical and Electronics Engineers, 2008, pp. 197–200. DOI: 10.1109/RFIC.2008.4561417.

[48] V. Vidojkovic et al. "A Low-voltage Folded-switching Mixer in 0.18-μ m CMOS". In: *IEEE Journal of Solid-State Circuits* 40.6 (2005), pp. 1259–1264. DOI: 10.1109/JSSC.2005.848034.

[49] Shuenn-Yuh Lee, Ming-Feng Huang, and C.J. Kuo. "Analysis and Implementation of a CMOS Even Harmonic Mixer with Current Reuse for Heterodyne/Direct Conversion Receivers". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 52.9 (2005), pp. 1741–1751. DOI: 10.1109/TCSI.2005.852487.

[50] Paul H. Young. *Electronic Communication Techniques*. Upper Saddle River, N.J. : Prentice Hall, cop., 2004.

[51] Stephen A. Maas. *Microwave mixers*. Artech House microwave library. Artech House Boston, London, 1993.

[52] Yongwang Ding and Ramesh Harjani. *High-linearity CMOS RF front-end Circuits*. New York : Springer, 2005.

[53] B. Gilbert. "A Precise Four-quadrant Multiplier with Subnanosecond Response". In: *Solid-State Circuits, IEEE Journal of* 3.4 (1968), pp. 365 –373. DOI: 10.1109/JSSC.1968.1049925.

[54] Behzad Razavi. *RF microelectronics*. Upper Saddle River, NJ : Prentice Hall, cop. 2012., 2012.

[55] H. Darabi and A.A. Abidi. "Noise in RF-CMOS mixers: a simple physical model". In: *IEEE Journal of Solid-State Circuits* 35.1 (2000), pp. 15–25. DOI: 10.1109/4.818916.

[56] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. Boston, MA: McGraw-Hill, 2001.

[57] Qiang Li et al. "CMOS RF mixer no-linearity design". In: *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems. MWSCAS 2001 (Cat. No.01CH37257)*. Institute of Electrical and Electronics Engineers, 2001, pp. 808–811. DOI: 10.1109/MWSCAS.2001.986310.

[58] Chien-Hsiung Feng et al. "Analysis of nonlinearities in RF CMOS amplifiers". In: *ICECS99. Proceedings of ICECS 99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357)*. Institute of Electrical and Electronics Engineers, 1999, pp. 137–140. DOI: 10.1109/ICECS.1999.812242.

[59] Chang-Hsi Wu and Yu-Po Lin. "Design of down-conversion mixer with gm-boosted stage combining low phase noise oscillator". In: *2010 International Symposium on Signals, Systems and Electronics*. Institute of Electrical and Electronics Engineers, 2010, pp. 1–4. DOI: 10.1109/ISSSE.2010.5607026.

[60] Chul Woo Byeon et al. "A 60 GHz Current-Reuse LO-Boosting Mixer in 90 nm CMOS". In: *IEEE Microwave and Wireless Components Letters* 22.3 (2012), pp. 135–137. DOI: 10.1109/LMWC.2011.2182636.

[61] Zhiqun Cheng et al. "An Improved Current Bleeding Mixer Based on CMOS Technology". In: *2010 International Conference on Communications and Mobile Computing*. Institute of Electrical and Electronics Engineers, 2010, pp. 128–131. DOI: 10.1109/CMC.2010.207.

[62] L.A. NacEachern and T. Manku. "A Charge-injection Method for Gilbert Cell Biasing". In: *Conference Proceedings. IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.98TH8341)*. Institute of Electrical and Electronics Engineers, 1998, pp. 365–368. DOI: 10.1109/CCECE.1998.682760.

[63] Cadence Design Systems, Inc. *SpectreRF User Guide*. Cadence Design Systems, Inc., 2003.

[64] Paul R. Gray. *Analysis and Design of Analog Integrated Circuits*. Chichester : John Wiley, cop., 2009.

[65] A. Hajimiri and T.H. Lee. "A general theory of phase noise in electrical oscillators". In: *Solid-State Circuits, IEEE Journal of* 33.2 (1998), pp. 179–194. DOI: 10.1109/4.658619.

[66] C McAndrew et al. "Correlated noise modeling and simulation". In: *Proc. Tech. Workshop Compact Modeling*. 2005, pp. 40–45.

[67] X. Li et al. *PSP 103.1*. http://pspmodel.asu.edu/downloads/psp103p1_summary.pdf retrieved on 2013-03-21. NXP Semiconductors, 2009.

[68] Cadence Design Systems, Inc. *Cadence Verilog-AMS Language Reference*. Cadence Design Systems, Inc., 2001.

[69] S.-G. Lee and J.-K. Choi. "Current-reuse bleeding mixer". In: *Electronics Letters* 36.8 (2000), p. 696. DOI: 10.1049/el:20000556.

[70] T. Melly et al. "An analysis of flicker noise rejection in low-power and low-voltage CMOS mixers". In: *Solid-State Circuits, IEEE Journal of* 36.1 (2001), pp. 102–109. DOI: 10.1109/4.896234.

[71] H. Kanaya et al. "Low flicker-noise and low leakage direct conversion CMOS mixer for 5GHz application". In: *2009 Asia Pacific Microwave Conference*. Institute of Electrical and Electronics Engineers, 2009, pp. 1631–1634. DOI: 10.1109/APMC.2009.5384348.

[72] L. Lemaitre et al. "Extensions to Verilog-A to support compact device modeling". In: *Proceedings of the 2003 IEEE International Workshop on Behavioral Modeling and Simulation*. Institute of Electrical and Electronics Engineers, 2003, pp. 134–138. DOI: 10.1109/BMAS.2003.1249872.

[73] Chao-Shiun Wang et al. "A 0.13 μm CMOS fully differential receiver with on-chip baluns for 60GHz broadband wireless communications". In: *2008 IEEE Custom Integrated Circuits Conference*. Institute of Electrical and Electronics Engineers, 2008, pp. 479–482. DOI: 10.1109/CICC.2008.4672125.

[74] Yang Shang et al. "An 8mW ultra low power 60GHz direct-conversion receiver with 55dB gain and 4.9dB noise figure in 65nm CMOS". In: *2012 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*. Institute of Electrical and Electronics Engineers, 2012, pp. 47–49. DOI: 10.1109/RFIT.2012.6401609.

[75] Behzad Razavi. "A mm-Wave CMOS Heterodyne Receiver with On-Chip LO and Divider". In: *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. Institute of Electrical and Electronics Engineers, 2007, pp. 188–596. DOI: 10.1109/ISSCC.2007.373357.

[76] Mikko Karkkainen et al. "60-GHz receiver and transmitter front-ends in 65-nm CMOS". In: *2009 IEEE MTT-S International Microwave Symposium Digest*. Institute of Electrical and Electronics Engineers, 2009, pp. 577–580. DOI: 10.1109/MWSYM.2009.5165762.

Code

Verilog-A

WFET_VS_Model/VS_Model/veriloga/veriloga.va

```
 // VerilogA for WFET, nw_int, veriloga

`include "constants.vams"
`include "disciplines.vams"

module NW_VS_Matrix(g,d,s);
inout g,d,s;
electrical g,d,s;

parameter real pi = 3.141528;
parameter real diam = 45e-9;
parameter real nNW = 192;
parameter real W = nNW*(diam*pi);
parameter real Vt = -0.11;
parameter real DIBL = 0.02;
parameter real tox = 7e-9;
parameter real L = 200e-9;
parameter real epsilon = 8.85e-12;
parameter real eox = 15;
//parameter real Cox = 2*pi*eox*epsilon*L/ln(1+(tox/r))*0.55;
//parameter real Cgg = nNW*Cox;
//parameter real Cinv = Cox/(W*L);
parameter real Wmin = 4.66e-9*eox/14.6;
parameter real Cinv = epsilon*eox/(tox+Wmin);
//parameter real Cmin = 2*pi*epsilon*eox*L/((2*(Wmin+tox)+diam)/diam);
parameter real Vmin = 0.026;
parameter real Vinj = 1.65e5;
parameter real my = 0.135;
parameter real m_elec = 9.1094e-31;
parameter real q = 1.602e-19;
parameter real Vdsats = Vinj*L/my;
parameter real alpha = 3.5;
parameter real beta = 1.8;
parameter real n = 3;
parameter real Rds = 30e3;
parameter real m_star = 0.2;
parameter real gamma = 2;


analog function real Id;
  input Vgs,Vds;
  real Vgs,Vds,Vgd,sgn,Vg,Vd,Vdsat,Veff,VeffA,VeffB,Ff,Qix,Fs,Ids,Rdsv,IdVS;
  begin
 sgn = 1;
       Vg = Vgs;
 if (Vds < 0) begin
  sgn = -1;
  Vg = Vgs-Vds;
 end
 Vd = abs(Vds);
    Veff = (Vg-Vt+DIBL*Vd);
 VeffA = Veff+alpha*Vmin/2;
```

```
 Ff = 1/(1+exp(VeffA/(alpha*Vmin)));
 VeffB = Veff+alpha*Vmin*Ff;
 Qix = Cinv*n*Vmin*ln(1+exp(VeffB/(n*Vmin)));
 Vdsat = Vdsats*(1-Ff)+Vmin*Ff;
 Fs = (Vd/Vdsat)/pow((1+pow((Vd/Vdsat),beta)),1/beta);
 Rdsv = Rds/nNW;
 if (Veff < 0) begin
  Rdsv = (Rds/nNW)*exp(-(Vg-Vt)/(12*Vmin));
 end
 Ids = Vd/Rdsv;
 IdVS = W*Qix*Vinj*Fs;
 Id = sgn*(IdVS+Ids);
end
endfunction

analog function real Cgs;
  input Vgs,Vds;
  real Vgs,Vds,sgn,Vg,Vd,Vdsat,Veff,VeffA,VeffB,Ff,Qix,Fs,m_e,k,k3,f1,f2,idVg,Cix,Cs,csk,
      Cfrac,Csmin,Cggmax;
  begin
        Vg = Vgs;
 if (Vds < 0) begin
  Vg = Vgs-Vds;
 end
 Vd = abs(Vds);
     Veff = (Vg-Vt+DIBL*Vd);
 VeffA = Veff+alpha*Vmin/2;
 Ff = 1/(1+exp(VeffA/(alpha*Vmin)));
 VeffB = Veff+alpha*Vmin*Ff;
 Qix = Cinv*n*Vmin*ln(1+exp(VeffB/(n*Vmin)));
 Vdsat = Vdsats*(1-Ff)+Vmin*Ff;
 Fs = (Vd/Vdsat)/pow((1+pow((Vd/Vdsat),beta)),1/beta);
 m_e = m_star*m_elec;
 if (L < 150e-9) begin
  m_e = (0.05+0.15*(max(20e-9,L)-20e-9)/(150e-9-20e-9))*m_elec;
 end
 k = 2*q*(Vd+Vmin/2)/(m_e*pow(Vinj,2));
        k3 = asinh(sqrt(k))/sqrt(k)-(sqrt(k+1)-1)/k;
 f1 = VeffB/(n*Vmin);
 f2 = VeffA/(alpha*Vmin);
 idVg = 1/(n*Vmin)*(1-exp(f2)/pow(1+exp(f2),2))*exp(f1);
 Cix = n*Vmin*idVg/(1+exp(f1));
 Cs = Cix*k3*Cinv*L*W;
 //Cgs = max(Cs,Cmin/2*(1-Fs*(1/3)));
 csk = 0.53-0.262*Vd+0.123*pow(Vd,2);
 Cfrac = 0.17;
 Csmin = Cfrac*(1-Fs*(1/3));
 Cggmax = Cinv*L*W*(0.87-Vd+0.53*pow(Vd,2));
 Cgs = Cs*(1-Cfrac/csk)+Csmin*Cggmax;
end
endfunction

analog function real Cgd;
  input Vgs,Vds;
  real Vgs,Vds,Vg,Vd,Vdsat,Veff,VeffA,VeffB,Ff,Qix,Fs,m_e,k,k4,f1,f2,idVg,Cix,Cd,cdk,Cfrac,
      Cdmin,Cggmax;
  begin
        Vg = Vgs;
```

```
 if (Vds < 0) begin
  Vg = Vgs-Vds;
 end
 Vd = abs(Vds);
     Veff = (Vg-Vt+DIBL*Vd);
 VeffA = Veff+alpha*Vmin/2;
 Ff = 1/(1+exp(VeffA/(alpha*Vmin)));
 VeffB = Veff+alpha*Vmin*Ff;
 Qix = Cinv*n*Vmin*ln(1+exp(VeffB/(n*Vmin)));
 Vdsat = Vdsats*(1-Ff)+Vmin*Ff;
 Fs = (Vd/Vdsat)/pow((1+pow((Vd/Vdsat),beta)),1/beta);
 m_e = m_star*m_elec;
 if (L < 150e-9) begin
  m_e = (0.05+0.15*(max(20e-9,L)-20e-9)/(150e-9-20e-9))*m_elec;
 end
 k = 2*q*(Vd+Vmin/2)/(m_e*pow(Vinj,2));
 k4 = (sqrt(k+1)-1)/k;
 f1 = VeffB/(n*Vmin);
 f2 = VeffA/(alpha*Vmin);
 idVg = 1/(n*Vmin)*(1-exp(f2)/(pow(1+exp(f2),2)))*exp(f1);
 Cix = n*Vmin*idVg/(1+exp(f1));
 Cd = Cix*k4*Cinv*L*W;
 //Cgd = max(Cd,Cmin/2*(1-Fs*(2/3)));
 cdk = 0.47-0.281*Vd+0.150*pow(Vd,2);
 Cfrac = 0.17;
 Cdmin = Cfrac*(1-Fs*(2/3));
 Cggmax = Cinv*L*W*(0.87-Vd+0.53*pow(Vd,2));
 Cgd = Cd*(1-Cfrac/cdk)+Cdmin*Cggmax;
end
endfunction

analog function real VdsSat;
  input Vgs,Vds;
  real Vgs,Vds,Vgd,sgn,Vg,Vd,Vdsat,Veff,VeffA,VeffB,Ff,Qix,Fs,Ids,Rdsv,IdVS;
  begin
 sgn = 1;
       Vg = Vgs;
 if (Vds < 0) begin
  sgn = -1;
  Vg = Vgs-Vds;
 end
 Vd = abs(Vds);
     Veff = (Vg-Vt+DIBL*Vd);
 VeffA = Veff+alpha*Vmin/2;
 Ff = 1/(1+exp(VeffA/(alpha*Vmin)));
 VeffB = Veff+alpha*Vmin*Ff;
 Qix = Cinv*n*Vmin*ln(1+exp(VeffB/(n*Vmin)));
 Vdsat = Vdsats*(1-Ff)+Vmin*Ff;
 Fs = (Vd/Vdsat)/pow((1+pow((Vd/Vdsat),beta)),1/beta);
 VdsSat = Vdsat * Fs;
end
endfunction

analog begin
  I(d,s)<+Id(V(g,s),V(d,s));

  // Add noise to the drain current
  // First add white noise (thermal), then flicker noise
```

```
  I(d,s)<+white_noise(gamma*4*`P_K*$temperature*(Id(V(g,s),V(d,s))/V(d,s)), "
      thermal_channel");
  I(d,s)<+flicker_noise(nNW*(diam*pi*1e6)*pow(10,(1.37*(V(g,s)-Vt)-15.26))*VdsSat(V(g,s), V
      (d,s)), 1, "flicker_channel");

  // Debug
  //$strobe("%M: nNW = %g, Id = %g mA, Vds = %g V, Vgs = %g V", nNW, I(d,s)*1000, V(d,s), V
      (g,s));
  //$strobe("%M: nNW = %g, Cgs = %g", nNW, Cgs(V(g,s),V(d,s)));

  I(g,s)<+Cgs(V(g,s),V(d,s))*ddt(V(g,s));
  I(g,d)<+Cgd(V(g,s),V(d,s))*ddt(V(g,d));

  //I(g,s)<+Cinv*L*W*ddt(V(g,s));
  //I(g,d)<+Cinv*L*W*ddt(V(g,d));
end

endmodule
```

OCEAN

all.il

```
;############################################################
;# #
;# MIXER TEST BENCH AUTOMATION #
;# Niklas Lindblad <tn08nl7@student.lth.se>, 2013 #
;# #
;############################################################

simulator( 'spectre )

; Erases content of any previous Waveform windows
clearAll();

; Change this if necessary
home_dir = "/home/piraten/atn08nli/"
src = strcat(home_dir "/master-thesis/src/cadence/tests");
results_dir = strcat(home_dir "/master-thesis/results");

; Spectre folders
spectre_design_dir = strcat(home_dir "/simulation/MixerTestBench/spectre/schematic/netlist/
    netlist");
spectre_results_dir = strcat(home_dir "/simulation/MixerTestBench/spectre/schematic");

; What frequencies to use?
load(strcat(src "/frequencies.il"));
center_frequency = (stop_frequency-start_frequency)/2 + start_frequency;

sprintf(center_frequency_s "%01.0f" center_frequency/1e9);
center_frequency_s = strcat(center_frequency_s "G");

sprintf(start_frequency_s "%01.1f" start_frequency/1e9);
start_frequency_s = strcat(start_frequency_s "G");
start_frequency_int = floor(start_frequency/1e9);

sprintf(stop_frequency_s "%01.1f" stop_frequency/1e9);
stop_frequency_s = strcat(stop_frequency_s "G");
stop_frequency_int = floor(stop_frequency/1e9);

; Load help functions and design variables
load(strcat(src "/functions.il"));
load(strcat(src "/variables.il"));

; Is there an active cell view containing a mixer?
if(uWindowIsActive() == nil
 exit();
)

; What are we simulating?
mixer_name = uGetMixerName();
current_time = uGetCurrentTime();
results_dir = strcat(results_dir "/" mixer_name "/" current_time);
uCreateDirectory(results_dir);

; Write parameters to file
fp = outfile( strcat(results_dir "/parameters.csv") "w");
```

```
fprintf(fp "Parameter\tValue\n");
foreach( v list("Lg" "Vinj" "NWx" "NWy" "dNW" "Vdd")
 fprintf(fp "%s\t%s\n", v, desVar(v));
)
; Save frequency data
fprintf(fp "%s\t%s\n", "start_frequency", start_frequency_s)
fprintf(fp "%s\t%s\n", "stop_frequency", stop_frequency_s)
close(fp);

; Save skip list
fp = outfile( strcat(results_dir "/skip.csv") "w");
fprintf(fp "%s\n", buildString( skip " "))
close(fp);

; Make sure the signal sources are set up correctly
;#################################################

; IF
uSetProperty("IF" "srcType" "dc");

; LO
uSetProperty("LO" "srcType" "sine");
uGetProperties("LO");
uSetProperty("LO" "fundname" "fLO");
uSetProperty("LO" "freq" "fLO");/net51
uSetProperty("LO" "vaDBm" "pLO");

; The RF source will change properties depending on
; what type of analysis we are doing.
procedure( uSetupRF( a )
 if(a == "dc"
  ; DC settings (for everything except
  ; linearity measurements)
  (progn
   uSetProperty("RF" "srcType" "dc");
   uSetProperty("RF" "vdc" "VRF");
         uSetProperty("RF" "pacDBm" "");
   uSetProperty("RF" "pacm" "1");
   uSetProperty("RF" "vaDBm" "");
         uSetProperty("RF" "smallSig" "TRUE");
  )
  (progn
   uSetProperty("RF" "srcType" "sine");
         uCreateProperty("RF" "fundname" "string" "fRF");
         uCreateProperty("RF" "freq" "string" "fRF");
         uCreateProperty("RF" "vaDBm" "string" "pRF");
       uSetProperty("RF" "vaDBm" "pRF");
         uCreateProperty("RF" "pacDBm" "string" "pRF");
         uSetProperty("RF" "pacDBm" "pRF");
   uSetProperty("RF" "pacm" "");
         uSetProperty("RF" "smallSig" "TRUE");
  )
 )
)

;#################################################
;# DC Power Consumption versus LO Signal Power #
;#################################################
```

```
if( (not (member "power" skip))
 (progn
 uSetupRF("dc");
 ocnWaveformTool( 'wavescan )
 simulator( 'spectre )
 design(spectre_design_dir)
 resultsDir(spectre_results_dir)
 analysis('pss ?fund center_frequency_s ?harms "10" ?errpreset "moderate"
  ?param "pLO" ?start "-10" ?stop "20" ?lin "10"
  ?restart "" )
 load(strcat(src "/variables.il"))
 desVar( "fLO" center_frequency_s)
 desVar( "fIF" "100M")
 desVar( "fRF" "(fLO+fIF)*1")
 desVar( "pLO" 7.5 )
 temp( 27 )
 save('all)
 run()
 results()
 selectResult( 'pss_td )
 current = i("CURRENT:in" ?result "pss_td");
 ocnPrint(?output strcat(results_dir "/current_vs_plo.dat") ?precision 16 ?numberNotation '
     scientific current);
 )
)
;#####################################################
;# Voltage Conversion Gain versus LO Signal Power #
;#####################################################
if( (not (member "gain" skip))
 (progn
 uSetupRF("dc");
 ocnWaveformTool( 'wavescan )
 simulator( 'spectre )
 design(spectre_design_dir);
 resultsDir(spectre_results_dir);
 analysis('pac ?start center_frequency_s ?stop "" ?maxsideband "2" )
 analysis('pss ?fund center_frequency_s ?harms "10" ?errpreset "moderate"
  ?param "pLO" ?start "-10" ?stop "15" ?lin "30"
  ?restart "" )
 load(strcat(src "/variables.il"))
 desVar( "fLO" center_frequency_s)
 desVar( "fIF" "100M")
 desVar( "fRF" center_frequency_s)
 desVar( "pLO" 7.5 )
 temp( 27 )
 save('all)
 run()
 results()
 selectResult( 'pac )
 conversionGain = db(harmonic(v("/net51" ?result "pac") '-1));
 ocnPrint(?output strcat(results_dir "/vcg_vs_plo.dat") ?precision 16 ?numberNotation '
     scientific conversionGain);
 if( doPlot
  plot(conversionGain);
 )
 )
)
;#####################################################
```

```
;# Voltage Conversion Gain versus RF Frequency #
;#################################################
if( (not (member "bandwidth" skip))
 (progn
 uSetupRF("dc");
 measurements_per_ghz = 2;
 for( f start_frequency_int*measurements_per_ghz stop_frequency_int*measurements_per_ghz
  ; Convert to string
  start_f = f * (1.0/measurements_per_ghz)
  stop_f = start_f + 0.01
  sprintf(start_f "%f" start_f)
  sprintf(stop_f "%f" stop_f)
  start_f = strcat(start_f "G")
  stop_f = strcat(stop_f "G")
  println(strcat(start_f " => " stop_f))

  ocnWaveformTool( 'wavescan )
  simulator( 'spectre )
  design(spectre_design_dir)
  resultsDir(spectre_results_dir)
  analysis('pac ?start start_f ?stop stop_f ?maxsideband "2" )
  analysis('pss ?fund start_f ?harms "10" ?errpreset "moderate" )
  load(strcat(src "/variables.il"))
  desVar( "pLO" 8.75 )
  desVar( "fLO" start_f )
  desVar( "fIF" "100M")
  desVar( "fRF" "(fLO+fIF)*1")
  temp( 27 )
  save('all)
  run()
  results()
  selectResult( 'pac )
  conversionGain = db(harmonic(v("/net51" ?result "pac") '-1));
  freq = 0
  sprintf(freq "%d" f)
  ocnPrint(?output strcat(results_dir "/vcg_vs_rf_" freq ".dat") ?precision 16 ?
      numberNotation 'scientific conversionGain);
 )
 )
)
;#################################################
;# NFdsb at 100 MHz versus LO power level #
;#################################################
if( (not (member "noise" skip))
 (progn
 uSetupRF("dc");
 ocnWaveformTool( 'wavescan )
 simulator( 'spectre )
 design(spectre_design_dir)
 resultsDir(spectre_results_dir)
 analysis('pnoise ?start "100M" ?stop "100M" ?maxsideband "10"
  ?oprobe "/IF" ?iprobe "/RF" ?refsideband "1" );
 analysis('pss ?fund center_frequency_s ?harms "10" ?errpreset "moderate"
  ?param "pLO" ?start "-5" ?stop "15" ?lin "30"
  ?restart "" )
 load(strcat(src "/variables.il"))
 desVar( "fLO" center_frequency_s)
 desVar( "fIF" "100M")
```

```
 desVar( "fRF" "(fLO+fIF)*1")
 desVar( "pLO" 7.5 )
 temp( 27 )
 save('all)
 run()

 ; NF DSB
 NFdsb=getData("NFdsb" ?result 'pnoise)
 ocnPrint(?output strcat(results_dir "/nfdsb_vs_plo.dat") ?precision 16 ?numberNotation '
     scientific NFdsb);
 if( doPlot
  plot(NFdsb);
 )
 )
)
;#####################################################
;# Port Isolation (RF to LO/IF and LO to IF/RF) #
;#####################################################
if( (not (member "isolation" skip))
 (progn
 uSetupRF("dc");

 ; Round up center frequency to make sure it is same as for other tests
 rounded_center_frequency = floor(center_frequency/1e9) + 1;
 sprintf(rounded_center_frequency_s "%d" rounded_center_frequency);
 rounded_center_frequency_s = strcat(rounded_center_frequency_s "G");

 sprintf(pac_frequency_s "%01.2f" rounded_center_frequency + 0.01);
 pac_frequency_s = strcat(pac_frequency_s "G");
 sprintf(pxf_frequency_s "%01.2f" rounded_center_frequency + 0.03);
 pxf_frequency_s = strcat(pxf_frequency_s "G");

 ocnWaveformTool( 'wavescan )
 simulator( 'spectre )
 design(spectre_design_dir)
 resultsDir(spectre_results_dir)
 analysis('pac ?start rounded_center_frequency_s ?stop pac_frequency_s ?maxsideband "2" )
 analysis('pxf ?p "/IF+" ?n "/gnd!" ?start rounded_center_frequency_s ?stop pxf_frequency_s
      ?maxsideband "2" )
 analysis('pss ?fund rounded_center_frequency_s ?harms "10" ?errpreset "moderate" )
 load(strcat(src "/variables.il"))
 uSetProperty("RF" "srcType" "sine");
 createNetlist();
 desVar( "pLO" 5 )
 desVar( "fLO" rounded_center_frequency_s )
 desVar( "fRF" rounded_center_frequency_s )
 temp( 27 )
 run()

 selectResult( 'pac )
 RF_to_LO = db(harmonic(v("/vLO+" ?result "pac") '-1));
 RF_to_IF = db(harmonic(v("/IF+" ?result "pac") '(0)));
 ocnPrint(?output strcat(results_dir "/rf_to_lo.dat") ?precision 16 ?numberNotation '
     scientific RF_to_LO);
 ocnPrint(?output strcat(results_dir "/rf_to_if.dat") ?precision 16 ?numberNotation '
     scientific RF_to_IF);

 selectResult( 'pxf )
```

```
 LO_to_IF = db(harmonic(getData("/LO" ?result "pxf") '(0)));
 LO_to_RF = db(harmonic(getData("/RF" ?result "pxf") '(0)));
 ocnPrint(?output strcat(results_dir "/lo_to_if.dat") ?precision 16 ?numberNotation '
     scientific LO_to_IF);
 ocnPrint(?output strcat(results_dir "/lo_to_rf.dat") ?precision 16 ?numberNotation '
     scientific LO_to_RF);
 )
)
;#####################################################
;# Linearity (1 dB compression and IIP3) #
;#####################################################
if( (not (member "linearity" skip))
 (progn
 uSetupRF("sine");

       ; Round up center frequency to make sure it is same as for other tests
       rounded_center_frequency = floor(center_frequency/1e9) + 1;
       sprintf(rounded_center_frequency_s "%d" rounded_center_frequency);
       rounded_center_frequency_s = strcat(rounded_center_frequency_s "G");

       sprintf(qpac_frequency_s "%01.4f" rounded_center_frequency + 0.11);
       qpac_frequency_s = strcat(qpac_frequency_s "G");

 sprintf(rf_frequency_s "%01.5f" rounded_center_frequency + 0.1);
 rf_frequency_s = strcat(rf_frequency_s "G");

 ocnWaveformTool( 'wavescan )
 simulator( 'spectre )
 design(spectre_design_dir)
 resultsDir(spectre_results_dir)
 analysis('qpac ?start qpac_frequency_s ?stop "" ?clockmaxharm "2" )
 analysis('qpss ?funds list("fLO" "fRF") ?maxharms list("5" "4") ?errpreset "moderate"
  ?param "pRF" ?start "-70" ?stop "5" ?lin "30"
  ?restart "" )
 load(strcat(src "/variables.il"))
 desVar( "fLO" rounded_center_frequency_s )
 desVar( "fRF" rf_frequency_s )
 temp( 27 )
 run()
 results()

 selectResult( 'qpss_fi );
 compressionPoint = compressionVRICurves((v("/net51" ?result "qpss_fi") - 0.0) '(-1 1) ?
     rport resultParam("IF:r" ?result "qpss_fi") ?epoint -70 ?gcomp 1);
 ocnPrint(?output strcat(results_dir "/1dB_compression.dat") ?precision 16 ?numberNotation
     'scientific compressionPoint);

 selectResult( 'qpac );
 IIP3 = ipnVRICurves((v("/net51" ?result "qpac") - 0.0) '(1 -2) '(-1 0) ?rport resultParam(
     "IF:r" ?result "qpac") ?epoint -60);
       ocnPrint(?output strcat(results_dir "/IIP3.dat") ?precision 16 ?numberNotation '
           scientific IIP3);
 )
)
;#####################################################
;# Post-simulation #
;#####################################################
;uNotify();
```

functions.il

```
;#####################################################
;# Help Functions #
;#####################################################

procedure( uGetInstance( n )
 return_inst = nil
 cv = geGetEditCellView()
       foreach(inst cv~>instances
              if(inst~>master then
                     unless(leIsContact(inst)
    if(inst~>name==n
                       return_inst = inst;
                      )
   )
              )
       )
 return_inst
)

procedure( uSetProperty( n p v )
 return_property = nil
 inst = uGetInstance(n)
 cv = geGetEditCellView()
 foreach(prop inst->prop
  if(prop->name == p
   (progn
    prop->value = v
    cdfparamCallback = prop~>callback
    errset( evalstring(cdfparamCallback) )
    schVIC(cv)
    dbSave(cv)
    return_property = prop;
   )
  )
 )
 return_property
)

procedure( uCreateProperty( n p type v )
 dbCreateProp(uGetInstance(n) p type v);
)

procedure( uPrintProperties( n )
       inst = uGetInstance(n)
       foreach( prop inst->prop
  printf("%s = %s\n", prop~>name, prop~>value);
       )
)

procedure( uGetProperties( n )
 inst = uGetInstance(n)
 foreach( prop inst->prop
  prop
 )
)
```

```
procedure( uGetMixerName( )
 mixer_name = nil
 cv = geGetEditCellView()
 rexCompile(".*Mixer.*")
       foreach(inst cv~>instances
              if(inst~>master then
                     unless(leIsContact(inst)
    if(rexExecute(inst~>cellName)
                       mixer_name = inst~>cellName;
                      )
  )
              )
       )
 mixer_name
)

procedure( uGetCurrentTime()
 id = ipcBeginProcess("date +%Y-%m-%d_%H_%M");
 ipcWait(id);
 timestamp = ipcReadProcess(id);
 timestamp = strncat("" timestamp strlen(timestamp)-1)
)

procedure( uGetCurrentTimeEpoch()
 id = ipcBeginProcess("date +%s");
 ipcWait(id);
 timestamp = ipcReadProcess(id);
 timestamp = strncat("" timestamp strlen(timestamp)-1)
)

procedure( uCreateDirectory(d)
 id = ipcBeginProcess(strcat("mkdir -p " d))
 ipcWait(id)
 ipcReadProcess(id)
)

procedure( uWindowIsActive( )
 window_active = t
 if(uGetMixerName() == nil
  window_active = nil
 )
 window_active
)

procedure( uNotify()
 id = ipcBeginProcess("bash /home/piraten/atn08nli/master-thesis/src/scripts/notify.sh")
 ipcWait(id)
 ipcReadProcess(id)
)
```

variables.il

```
load(strcat(src "/variables/NWFET_60nm.il"));
load(strcat(src "/variables/mixerA.il"));
load(strcat(src "/variables/testbench.il"));
```

variablesmixerA.il

```
desVar( "I_tail" 5.8m)
desVar( "Vdd" 1.5 )
desVar( "R" 270 )
desVar( "xRF" 6 )
desVar( "xLO" 6 )
;desVar( "cf" "1/(2*3.14*fIF*R)")
desVar( "cf" "1/(2*3.14*3*fIF*R)")
```

variablesNWFET_60nm.il

```
; Design variables for 60 nm gate length NWFET

desVar( "PadW" "(NWy+2)*NW_Spacing" );
desVar( "nRF" "NWx*NWy" );
desVar( "PadW_LO" "(NWy*xLO+2)*NW_Spacing" );
desVar( "Rg" "5.6e-8*PadL/2/(0.060u*PadW)" );
desVar( "PadL" "(NWx+1)*NW_Spacing" )
desVar( "Rds" "xR*9e3" )
desVar( "Vt" -0.11 )
desVar( "Lg" 60n )
desVar( "Rdi" 1000 )
desVar( "Rsi" 460 )
desVar( "diam" 45e-9 )
desVar( "tox" 7n )
desVar( "Rd_e" 500m )
; NWy must be kept at 8
desVar( "NWy" 8 )
desVar( "NWx" 6 )
; Standard 48 wires
desVar( "nNW" "NWx*NWy")
desVar( "CgdiN" 9.2a )
desVar( "CdseN" 0 )
desVar( "CdseC" 106a )
desVar( "CgdeN" 9.4a )
desVar( "CgdeC" 238a )
desVar( "CgsiN" 10.1a )
desVar( "CgseN" 11a )
desVar( "CgseC" 321a )
desVar( "Perm_HfO2" 15)
desVar( "Perm_S18xx" 3)
desVar( "Perm_SiO2" 3.9)
desVar( "Perm_Si3N4" 7)
desVar( "Rsheet" 6 )
desVar( "DIBL" 0.07 )
desVar( "Vinj" 1.95e5 )
desVar( "SS" 3 )
desVar( "my" 0.13 )
desVar( "dNW" 45e-9 )
desVar( "NW_Spacing" 60n )
desVar( "Vt" -0.11 )
desVar( "xR" 20 )
```

variablestestbench.il

```
desVar( "pLO" 8 )
desVar( "vRF" 20m )
desVar( "pRF" -50 )
;desVar("xVod" 2 )
desVar( "fRF" 60G )
desVar( "vLO" "Vdd/2" )
desVar( "VRF" -185m )
desVar( "fIF" 100M )
desVar( "Vod" -60m )
desVar( "fLO" 60G )
desVar( "VLO" "VRF+85m" )
desVar( "VRF" -185m )
```

MatLab

analyze.m

```
function [ output_args ] = analyze( result )
    clc;
    close all;

    lineWidth = 4;
    markerSize = 10;

    % Change default axes fonts.
    set(0,'DefaultAxesFontName', 'Latin Modern')
    set(0,'DefaultAxesFontSize', 20)

    % Change default text fonts.
    set(0,'DefaultTextFontname', 'Latin Modern')
    set(0,'DefaultTextFontSize', 20)

    results_dir = [pwd '/../../results/' result];
    summary = fopen([results_dir '/summary.csv'],'wt');

    %% Load parameters
    p = importdata([results_dir '/parameters.csv']);
    vdd = 0;
    L_g = 0;
    for i = 1:length(p.textdata)-1
       if strcmp(p.textdata(i, 1), 'Vdd')
           vdd = p.data(i-1);
            fprintf(summary, 'VDD_V\t%s\n', num2str(vdd));
       end
       if strcmp(p.textdata(i, 1), 'Lg')
           L_g = p.data(i-1);
           fprintf(summary, 'Gate_Length_nm\t%s\n', num2str(L_g/1e-9));
       end
    end

    %% Load skip list
    skip = importdata([results_dir '/skip.csv']);
    if ~isempty(skip)
        skip = regexp(skip,' ', 'split');
    else
        skip = {'none'};
    end

    %% Voltage Conversion Gain vs. LO Power
    if ~strcmp('gain', skip{1})
        d = importdata([results_dir '/vcg_vs_plo.dat']);
        pLO = d.data(1:end, 1);
        gain = d.data(1:end, 2);
        figure(1);
        plot(pLO, gain, 'k-O', 'LineWidth', lineWidth, 'MarkerSize', markerSize);
        [value, index] = max(gain);
        max_gain_pLO = pLO(index);
        title(['Gain vs. LO Power (L_g = ' num2str(L_g/1e-9) ' nm) Max. ' sprintf('%5.2f',
            value) ' dB @ ' num2str(max_gain_pLO) ' dBm']);
        xlabel('LO Power [dBm]');
        ylabel('Conversion Gain [dB]');
```

```
    grid on;

    % Write to summary
    fprintf(summary, 'Conversion_Gain_dB\t%s\n', sprintf('%5.2f', value));
    fprintf(summary, 'pLO_dBm\t%s\n', num2str(max_gain_pLO));
end
%% Voltage Conversion Gain vs. RF Frequency
if ~strcmp('bandwidth', skip{1})
    rf_freq = [];
    rf_gain = [];
    rf_measurements = dir([results_dir '/vcg_vs_rf_*.dat']);
    for i = 1:length(rf_measurements)-1;
        m = rf_measurements(i).name;
        [n s e] = regexp(m, '[0-9]', 'match', 'start', 'end');
        o = substr(m, s(1), length(s));
        rf_ghz = str2num(o) * 0.5;
        rf = importdata([results_dir '/' m]);
        for j = 2:3
            f = rf.data(j, 1) * 10e-9 + rf_ghz;
            g = rf.data(j, 2);
            rf_freq = [rf_freq f];
            rf_gain = [rf_gain g];
        end
    end
    figure(2);
    plot(rf_freq, rf_gain, 'ko', 'LineWidth',2);
    title(['Gain vs. RF Frequency (L_g = ' num2str(L_g/1e-9) ' nm)']);
    xlabel('RF Frequency [GHz]');
    ylabel('Conversion Gain [dB]');
    axis([min(rf_freq) max(rf_freq) min(rf_gain) max(rf_gain)]);
    fprintf(summary, 'Conversion_Gain_Difference_dB\t%s\n', abs(min(rf_gain)-max(
        rf_gain)));
end
%% Noise vs. LO power
if ~strcmp('noise', skip{1})
    d = importdata([results_dir '/nfdsb_vs_plo.dat']);
    pLO = d.data(1:end, 1);
    NF = d.data(1:end, 2);
    figure(4);
    plot(pLO, NF, 'k-O', 'LineWidth', lineWidth, 'MarkerSize', markerSize);
    grid on;
    title(['Noise Figure (DSB) vs. LO Power (L_g = ' num2str(L_g/1e-9) ')'] );
    xlabel('LO Power [dBm]');
    ylabel('NF_{DSB} [dB]');
    % Write to summary
    %fprintf(summary, 'Conversion_Gain_dB\t%s\n', sprintf('%5.2f', value));
    %fprintf(summary, 'pLO_dBm\t%s\n', num2str(max_gain_pLO));
end
%% Power Consumption
if ~strcmp('power', skip{1})
    fid = fopen([results_dir '/current_vs_plo.dat']);
    tline = fgets(fid);
    pLO = NaN;
    I = [];
    t = [];
    final_pLO = [];
    final_I = [];
    while ischar(tline)
```

```
        [token, remain] = strtok(tline);
        if strcmp(token, 'pLO')
           I_avg = mean(I);
           if ~isnan(I_avg)
            final_pLO = [final_pLO pLO];
            final_I = [final_I I_avg];
           end
        pLO = str2num(substr(remain, 3));
        I = [];
        t = [];
        elseif strcmp(token, 'time')

        else
            t = [t str2num(token)];
            I = [I abs(str2num(remain))];
        end
    tline = fgets(fid);
    end
    fclose(fid);

    power = final_I .* vdd .* 1000;
    figure(3);
    plot(final_pLO, power, 'kx', 'LineWidth', lineWidth, 'MarkerSize', markerSize * 2);
    title(['Power Consumption vs. LO Power (L_g = ' num2str(L_g/1e-9) ' nm, VDD = '
        num2str(vdd) ' V)']);
    xlabel('LO Power [dBm]');
    ylabel('Power Consumption [mW]');
    grid on;
    fprintf(summary, 'Power_Consumption_mW\t%s\n', num2str(power(find(final_pLO == 8)))
        );
end
 %% Linearity
if ~strcmp('linearity', skip{1})

    % 1 dB compression point
    d = importdata([results_dir '/1dB_compression.dat'])
    pRF = d.data(1:end, 1);
    out = d.data(1:end, 2);
    p = polyfit(pRF(1:5), out(1:5), 1);
    outFit = polyval(p, pRF);
    difference = out - outFit;
    [c index] = min(abs(difference+1));
    closestValue = difference(index);
    inputReferred_1dB_Compression = c;
    fprintf(summary, 'inputReferred_1dB_Compression_dBm\t%s\n', num2str(
        inputReferred_1dB_Compression));

    % IIP3
    d = importdata([results_dir '/IIP3.dat']);
    pRF = d.data(1:end, 1);
    firstOrder = d.data(1:end, 3);
    firstOrderFit = polyfit(pRF(1:5), firstOrder(1:5), 1);
    thirdOrder = d.data(1:end, 2);
    thirdOrderFit = polyfit(pRF(1:5), thirdOrder(1:5), 1);

    IIP3 = (thirdOrderFit(2) - firstOrderFit(2))/(firstOrderFit(1) - thirdOrderFit(1));
    OIP3 = thirdOrderFit(2) + thirdOrderFit(1) * IIP3;
    fprintf(summary, 'IIP3_dBm\t%s\n', num2str(IIP3));
```

```
        fprintf(summary, 'OIP3_dBm\t%s\n', num2str(OIP3));
    end
    %% Port isolation
    if ~strcmp('isolation', skip{1})

        fid = fopen([results_dir '/port_isolation.csv'],'wt');

        % LO to IF
        d = importdata([results_dir '/lo_to_if.dat']);
        LO_to_IF = sprintf('%.2f', mean(d.data(1:end,2)));
        fprintf(fid, 'LO_to_IF_dB\t%s\n', LO_to_IF);
        fprintf(summary, 'LO_to_IF_dB\t%s\n', LO_to_IF);

        % LO to RF
        d = importdata([results_dir '/lo_to_rf.dat']);
        LO_to_RF = sprintf('%.2f', mean(d.data(1:end,2)));
        fprintf(fid, 'LO_to_RF_dB\t%s\n', LO_to_RF);
        fprintf(summary, 'LO_to_RF_dB\t%s\n', LO_to_RF);

        % RF to IF
        d = importdata([results_dir '/rf_to_if.dat']);
        RF_to_IF = sprintf('%.2f', mean(d.data(1:end,2)));
        fprintf(fid, 'RF_to_IF_dB\t%s\n', RF_to_IF);
        fprintf(summary, 'RF_to_IF_dB\t%s\n', RF_to_IF);

        % RF to LO
        d = importdata([results_dir '/rf_to_lo.dat']);
        RF_to_LO = sprintf('%.2f', mean(d.data(1:end,2)));
        fprintf(fid, 'RF_to_LO_dB\t%s\n', RF_to_LO);
        fprintf(summary, 'RF_to_LO_dB\t%s\n', RF_to_LO);

        fclose(fid);
    end
    %% Save all figures
    SaveAllFigures(results_dir, '', 'pdf')
    SaveAllFigures(results_dir, '', 'png')

    %% Final clean up
    fclose(summary);
end
```

nf2f.m

```
function [ f ] = nf2f( nf )
% Convert NF to F
f = 10.^(nf/10);
end
```

LNA_design.m

```
function [ h ] = LNA_design( Mixer_NF, Mixer_VCG )
%LNA_DESIGN Plot LNA requirements for given mixer NF and VCG

[G,NF] = meshgrid([30-Mixer_VCG:-0.5:20-Mixer_VCG], [0.5:0.1:3]);
Z = 10.*log10(nf2f(NF) + (nf2f(Mixer_NF)-1)./10.^(G/20));

contourf(G, NF, Z);
set(gca,'XDir','Reverse')
colormap jet;
```

```
cb = colorbar('location', 'southoutside');
title(['LNA Requirements Mixer NF = ' num2str(Mixer_NF) ' dB & Gain = ' num2str(Mixer_VCG)
    ' dB']);
xlabel('LNA Gain [dB]');
ylabel('LNA NF [dB]');
set(get(cb,'xlabel'),'String', 'Receiver NF [dB]');
h = Z;
end
```

Schematics

Test Bench Implementation

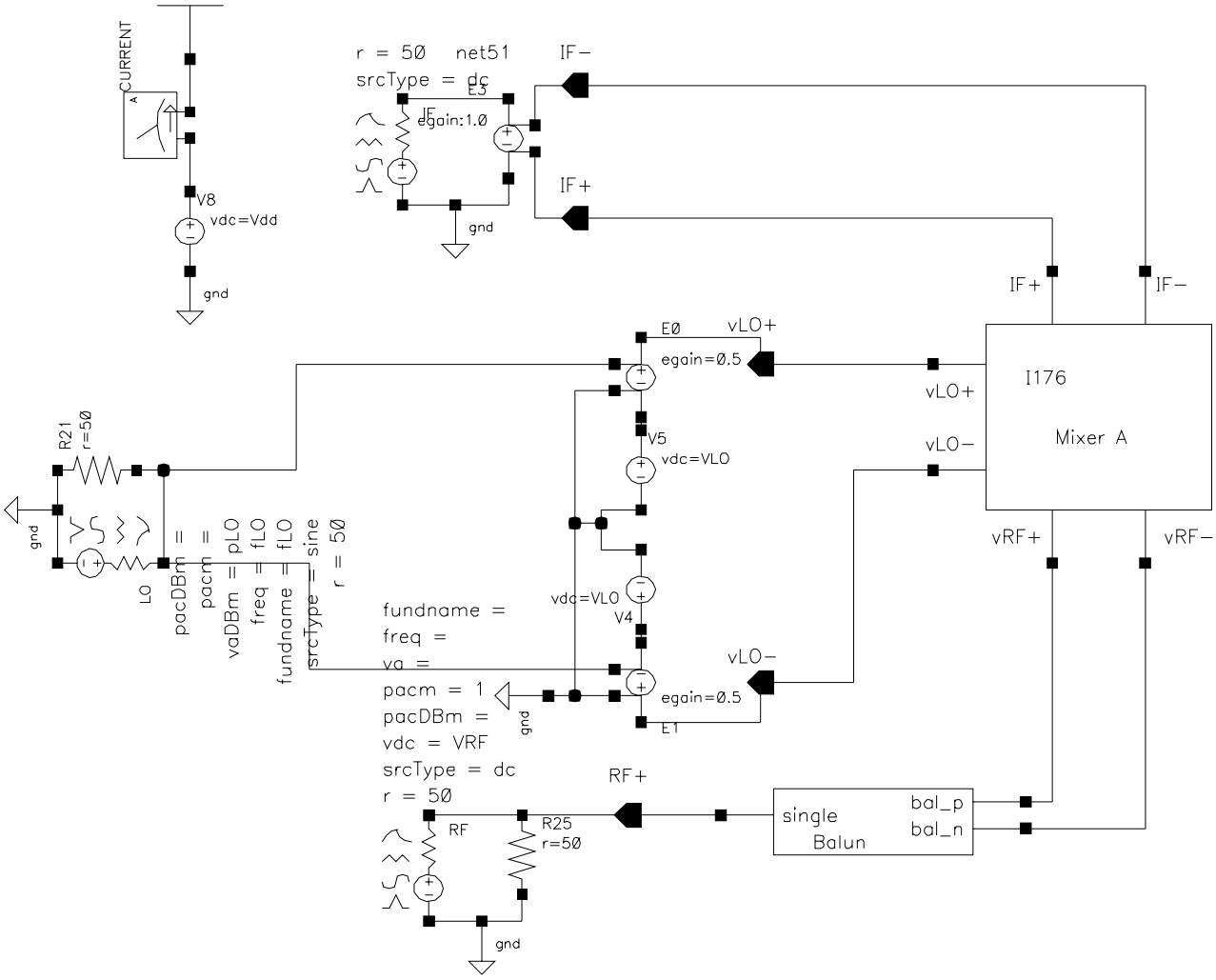

Figure 45: Schematic for the test bench setup.

Mixer A

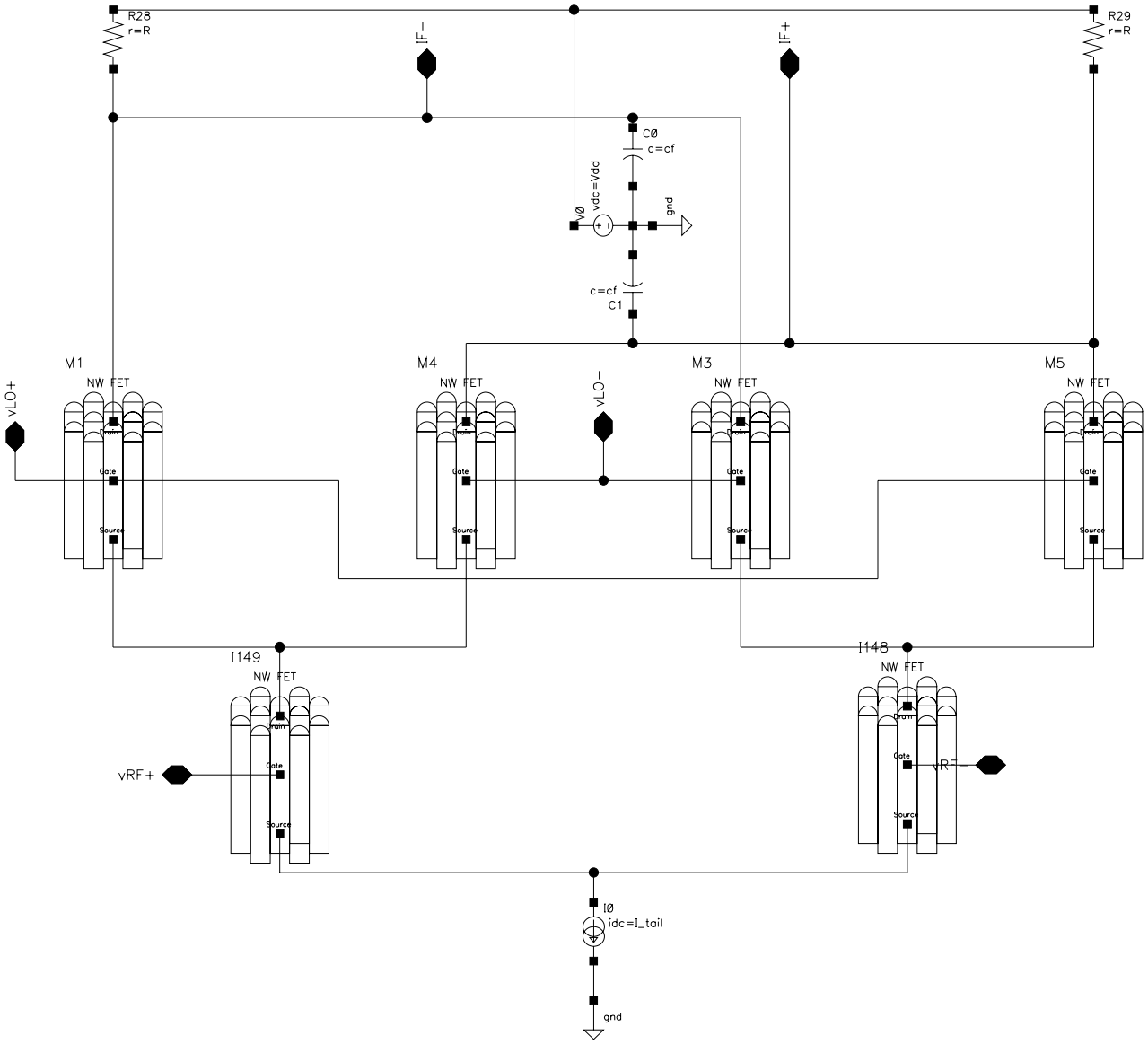

Figure 46: Schematic for mixer A.

Mixer B

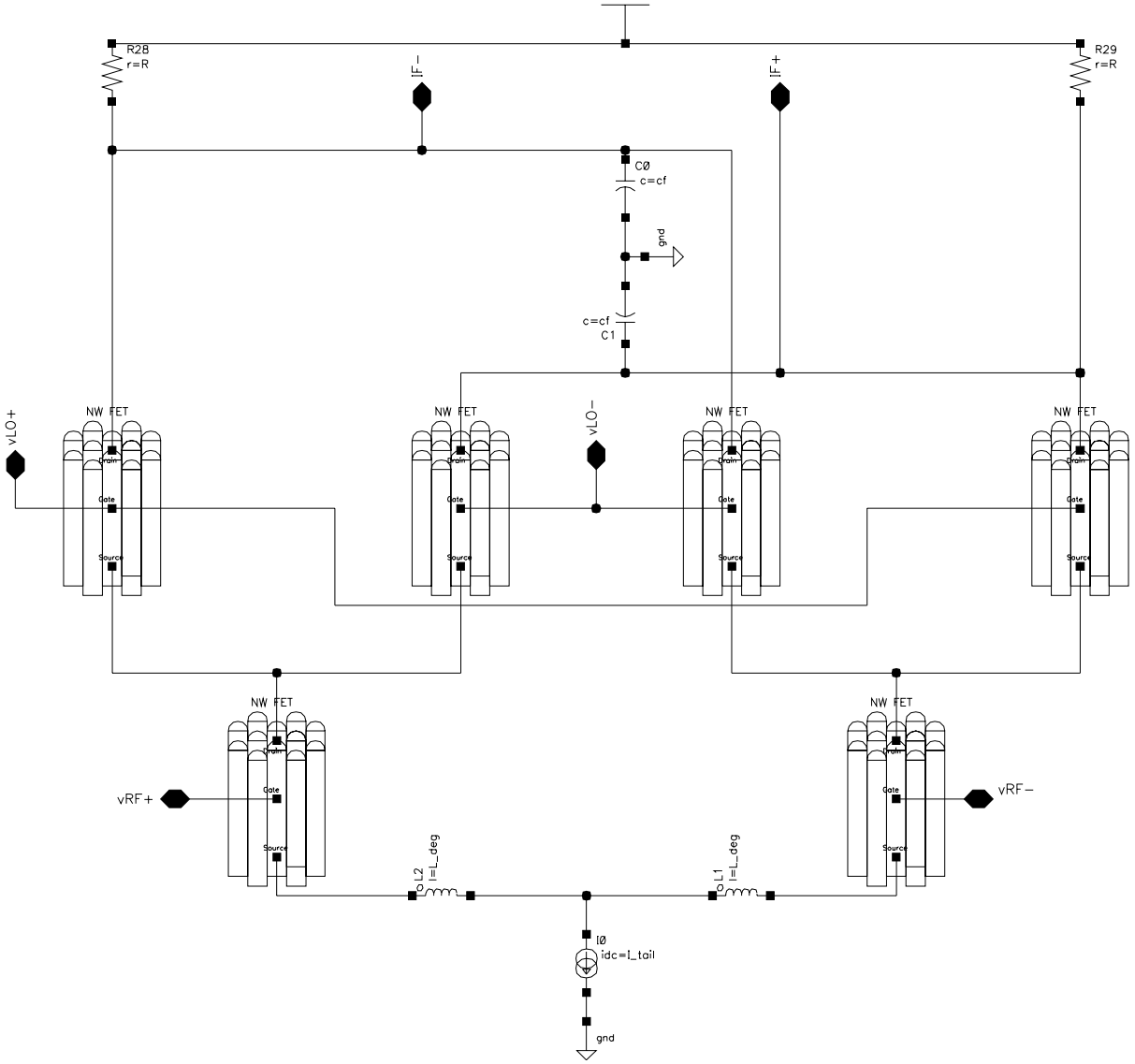

Figure 47: Schematic for mixer B.

Mixer C

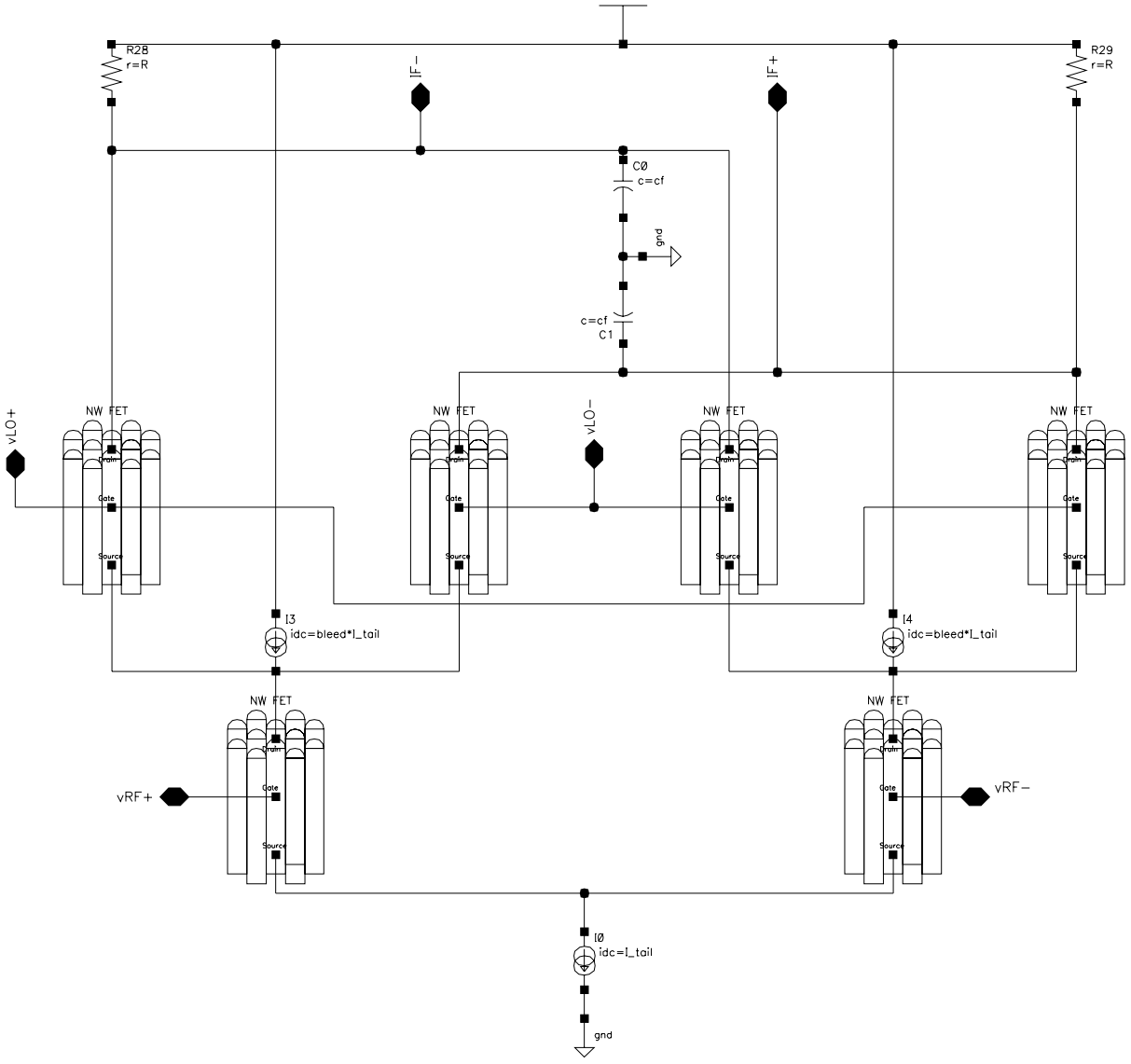

Figure 48: Schematic for mixer C.

List of Abbreviations

DSB Dual Sideband.

HBTs Heterojunction Bipolar Transistors.

IF Intermediate Frequency.
IIP3 Input-referred Third-order Intercept Point.
InAs Indium Arsenide.
InP Indium Phosphide.

LNA Low Noise Amplifier.
LO Local Oscillator.

mmWave Millimetre Wave.
MOSFETs Metal-Oxide Semiconductor Field-Effect Transistors.

NF Noise Figure.
NR Noise Ratio.
NWFETs Nanowire Field-Effect Transistors.

OCEAN Open Command Environment for Analysis.
OIP3 Output-referred Third-order Intercept Point.

PAC Periodic Alternating Current.
Pnoise Periodic Noise.
PSS Periodic Steady-State.
PXF Periodic Transfer Function.

QPAC Quasi-Periodic Alternating Current.
QPSS Quasi-Periodic Steady-State.

RF Radio Frequency.

SiGe Silicon-Germanium.
SOI Silicon On Insulator.
SSB Single Sideband.

ULSI Ultra-Large Scale Integration.

VCO Voltage Controlled Oscillator.