

Stretch Processing Radar RFIC System Analysis and Front-end Design

by

Zachary Hubbard

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 7, 2012

Keywords: stretch processing, chirp, LFM, group delay, radar, LNA

Copyright 2012 by Zachary Hubbard

Approved by

Fa Foster Dai, Chair, Professor of Electrical and Computer Engineering
Michael Baginski, Associate Professor of Electrical and Computer Engineering
Stuart Wentworth, Associate Professor of Electrical and Computer Engineering

Abstract

Current 0.13 μm SiGe BiCMOS technology allows for most of the components for an X-band radar to be fabricated on a single IC. This paper discusses system design and analysis of a stretch processing radar. The design uses a single ADC, no matched filters, and wide-band waveforms. Chapters include an in-depth analysis of stretch processing, LNA design concepts, and test software design for the IC.

Table of Contents

Abstract	ii
List of Figures	vi
List of Tables	x
List of Abbreviations	xi
1 Introduction	1
1.1 Integrated Radar	2
1.2 Radar Overview	3
1.2.1 Pulsed Radar	3
1.2.2 Pulse Compression	6
1.2.3 Stretch Processing	8
2 System	12
2.1 Simulation	12
2.1.1 Transmitter	12
2.1.2 Receiver	17
2.2 Cadence Simulation	23
2.2.1 TX Simulation	24
2.2.2 RX Simulation	25
3 Stretch Processing	33
3.1 Pulsewidth and LFM Start Frequency	33
3.1.1 Pulsewidth	33
3.1.2 Offset Frequency	34
3.1.3 Range Window	34
3.2 Doppler	36

3.3	Group Delay	38
3.3.1	GD Variation	40
3.3.2	GD Simulation	41
3.3.3	GD Power Loss	43
3.3.4	GD Summary	45
3.3.5	GD Range Resolution Effect	46
3.4	IQ Imbalance	47
3.4.1	Phasor Representation	49
3.4.2	Linear Combination	49
3.5	Processing Gain	52
3.5.1	Matched Filter	53
3.5.2	Processing Gain Derivation	53
3.5.3	ROC Processing Gain	57
3.6	Correct Stretch Processing on ROC	59
3.6.1	Analysis	60
3.7	Interference	64
4	Circuits	66
4.1	Group Delay Compensation	66
4.1.1	Equalizer	67
4.1.2	LNA with Equalizer	69
4.2	Common-Base Amplifier	75
4.2.1	CB-CC Simulation Results	79
4.2.2	Integrated Duplexer	83
5	Software	89
5.1	Test Setup	89
5.2	Software Needs	90
5.3	Platform Choice	91

5.4	Code Structure	92
5.4.1	GUI	92
5.4.2	ROCDriver	93
5.4.3	Serial Communication	95
5.4.4	Load and Save ROC States	97
5.4.5	Tools Tab	97
5.4.6	ADC Tab	99
5.4.7	Packaging	100
6	Summary	101
6.1	Simulation	101
6.2	Stretch Processing	101
6.2.1	Analysis	102
6.3	Circuits	105
6.4	Software	106
6.5	Future Work	106
	Bibliography	108

List of Figures

1.1	PRI	4
1.2	Two different waveforms with equal average power	5
1.3	Frequency vs. time of an LFM waveform	8
1.4	a) Time domain of LFM, b) Frequency domain of LFM	9
1.5	Stretch Processing	10
2.1	ROC block diagram	13
2.2	LFM waveform at IF	15
2.3	Single-sideband LFM waveform at IF	16
2.4	Filtered RF waveform in the frequency domain	17
2.5	Received BB waveform in the frequency domain	19
2.6	Destretching with a return at $\pm 3\mu s$	20
2.7	Example of a received waveform after destretching	21
2.8	Destretched waveform return mixed with the offset frequency	23
2.9	Result of subtracting the two channels	24
2.10	45 MHz transmitter simulation result	26

2.11	120 MHz transmitter simulation result	27
2.12	195 MHz transmitter simulation result	28
2.13	RX IF VGA output	29
2.14	RX simulation results for low gain	30
2.15	RX simulation results for low gain	31
3.1	LFM with frequency offset	35
3.2	GD Verification	43
3.3	The effect of Δ GD on range resolution	48
3.4	Contour plot for the power of the image frequency	52
3.5	Plot showing chirp-squared term	58
3.6	a)Chirp-squared term hiding a smaller tone b)Ideal stretch Processing	59
3.7	Updated Stretch Processing	61
4.1	Fifth order Butterworth LPF with GD response	66
4.2	Equalizer	67
4.3	Equalizer plots a) impedance b) phase c) group delay	68
4.4	Cascode LNA	70
4.5	LNA gain with no equalizer	72
4.6	LNA gain with equalizer	73

4.7	LNA NF	74
4.8	a) S_{22} with no equalizer b) S_{22} with equalizer	74
4.9	Common-base amplifier with voltage-voltage feedback	76
4.10	NF_{min} vs. I_{EE} for determining J_c	77
4.11	NF_{min} vs. emitter length	77
4.12	The interface between CB and CC	78
4.13	Schematic of CB-CC LNA	80
4.14	S_{11} and S_{22} are simultaneously matched	81
4.15	Input impedance of CB-CC	82
4.16	NF and NFmin comparison	82
4.17	CB-CC Gain	83
4.18	Autotransformer used as hybrid transformer	84
4.19	Duplexer	85
4.20	S-parameters for LNA with duplexer	86
4.21	Duplexer isolation	87
4.22	Duplexer and LNA NF	88
5.1	SPI diagram	89
5.2	ROC communication diagram	90

5.3 ROC Interface SPI tab 95

5.4 ROC Interface communication tab 96

5.5 ROC Interface Excel tab 98

5.6 ROC Interface ADC tab 99

6.1 Updated Stretch Processing 103

List of Tables

2.1	Delay and frequency values for RX simulation	32
3.1	Varying α to adjust the range window with constant range resolution	36
3.2	GD Simulation Parameters	42
4.1	CB-CC performance summary	83

List of Abbreviations

BJT	bipolar junction transistor
CW	continuous wave
GD	group delay
HBT	heterojunction bipolar transistor
IF	intermediate frequency
LFM	linear frequency modulation
LO	local oscillator
LPF	low pass filter
PLL	phase-lock loop
PRF	pulse repetition frequency
PRI	pulse repetition interval
ROC	radar-on-chip
SPI	serial peripheral interface

Chapter 1

Introduction

As transistor technologies continue to improve and decrease feature size, fabricating entire transceivers on a single chip is possible. Some radios in the cellular or GPS range can be fabricated on a single chip using high performance CMOS technology. High frequency analog applications are still more suitable to bipolar junction transistors (BJTs) . Traditionally, BJTs are optimal for high frequency analog operations because of their high speed, high gain, and low noise. CMOS transistors, on the other hand, are suited for digital circuits because of their low power consumption, high input impedance, and larger noise.

As CMOS technologies have improved, they have taken the place of BJTs in some analog circuitry. When possible, mixed-signal circuits fabricated entirely in CMOS is desirable because of the price. For mixed signal ICs that require BJT performance, BiCMOS is the solution. By using several extra steps in the fabrication, BJT transistors are available for use [7].

The transition frequency, f_T is a measure of the high frequency capabilities of a transistor. It is the frequency at which the short-circuit gain of the transistor is equal to unity. In practice, amplifiers can be fabricated for operating frequencies of f_T divided by ten or fifteen. The transition frequency is limited by the base charging capacitance and the emitter-base depletion layer capacitance. A large portion of this capacitance is determined by the base transit time. The base transit time can be reduced by reducing the size of the base. This is one reason that transistor scaling increases performance. Transistor scaling has performance limits though. As the base width is reduced, the base resistance increases. When the base resistance increases enough, a time constant forms that limits frequency performance.

Silicon BJT technology would have reached its peak if not for heterojunction devices. A heterojunction is a pn junction made of two different materials. SiGe devices use SiGe in the base of a BJT to increase device performance. Without going into too much detail, heterojunction bipolar transistors (HBTs) work because of the different band gaps of the two materials. The main benefit of SiGe HBTs is that the base width can be reduced while the base resistance remains constant. Therefore, the maximum operating frequency is increased. SiGe HBTs also have better noise performance than comparable BJTs. Another big advantage of SiGe HBTs over III-V materials such as GaAs is that they can be fabricated with a BiCMOS process. The system discussed in this paper was fabricated using a 0.13 μm SiGe BiCMOS technology.

1.1 Integrated Radar

The mixed signal capabilities afforded by SiGe HBT BiCMOS technology allows for most of the components of an X-band (8-12 GHz) radar to be fabricated on a single IC. The only components that cannot be fabricated on the IC practically are the IF filters, power amplifier, and antenna. An integrated radar has many advantages. The cost and mass producibility alone make it a worthy endeavor. An integrated radar would also weigh less and consume less power than otherwise.

Another common radar component that is hard to integrate for many waveforms is the matched filter. A matched filter correlates a received signal with the waveform that was transmitted. A matched filter has an optimum SNR. Matched filters are usually implemented with SAW filters or in digital processing. They work only for the specific waveform that they are designed for. If the radar uses multiple waveforms, a bank of matched filters is needed. A bank of custom designed off-chip components is not good for an integrated radar. Implementing a matched filter digitally also has a disadvantage. The bandwidth of the waveform is limited to the capabilities of the ADC. To overcome the difficulties of a matched filter, an alternative signal processing method is used. Stretch processing was first suggested

by W.J. Caputi in 1971 [4]. Stretch processing can be implemented on the IC and reduces the required bandwidth of the ADC. Much of this thesis has to do with implementation and analysis of stretch processing.

1.2 Radar Overview

An introduction to radar is prerequisite to fully explain the following radar system analysis. A comprehensive introduction is not needed. Only the basic principles to establish the foundation for the rest of the thesis are given. The introduction will focus on pulsed radar waveforms.

Radar waveforms can be broadly classified as either pulsed or continuous wave (CW). A CW radar continuously transmits a signal. A pulsed radar emits a signal for a time, usually on the order of μs . It then quits transmitting and “listens” for the return signal for a set time before transmitting again. A CW radar is normally bistatic. Bistatic means that there are separate antennas for receiving and transmitting the waveform. Bistatic also implies that the two antennas are physically separated. This is usually necessary in a CW radar to avoid the high-power transmitter interfering or damaging the sensitive receiver. The alternative to a bistatic configuration is monostatic configuration. Monostatic radars are pulsed. A monostatic radar shares the same antenna for receiving and transmitting. When the radar is transmitting, the receiver is usually turned off. When the radar is receiving or “listening,” the transmitter is turned off. This document will focus on a pulsed monostatic configuration.

1.2.1 Pulsed Radar

The rate at which a radar pulses is the pulse repetition frequency (PRF). The PRF is measured in Hz. The period of the PRF is the pulse repetition interval (PRI), measured in seconds. Figure 1.1 shows the PRI of a pulsed sine wave. The duty cycle of the pulse is the fraction of the PRI that the wave is transmitting. The equation for duty cycle is

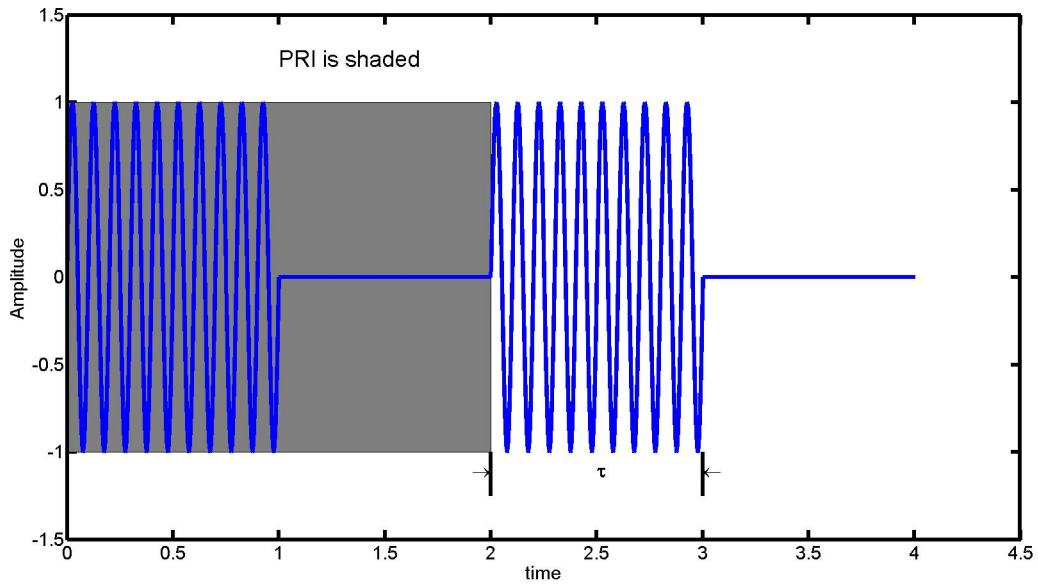


Figure 1.1: PRI

$$d_t = \frac{\tau}{PRI} = \tau \cdot PRF \quad (1.1)$$

The average power of a radar pulse is important. The average power determines the strength of a received signal. The average power has to increase as the target reflects less power or the desired target range is increased. It is calculated as the product of the duty cycle and the peak power, P_t :

$$P_{avg} = P_t \cdot d_t \quad (1.2)$$

The average power can be boosted by increasing the peak power or increasing τ . For illustration, Figure 1.2 demonstrates two waveforms with equal energy content and different τ s and peak powers. Often hardware limitations create the ceiling for the peak power. There are also problems associated with extending the transmitting time of the pulse. The minimum detectable range of the radar system is determined by the length of τ because the radar cannot start receiving until it has stopped transmitting. A target cannot be detected if its

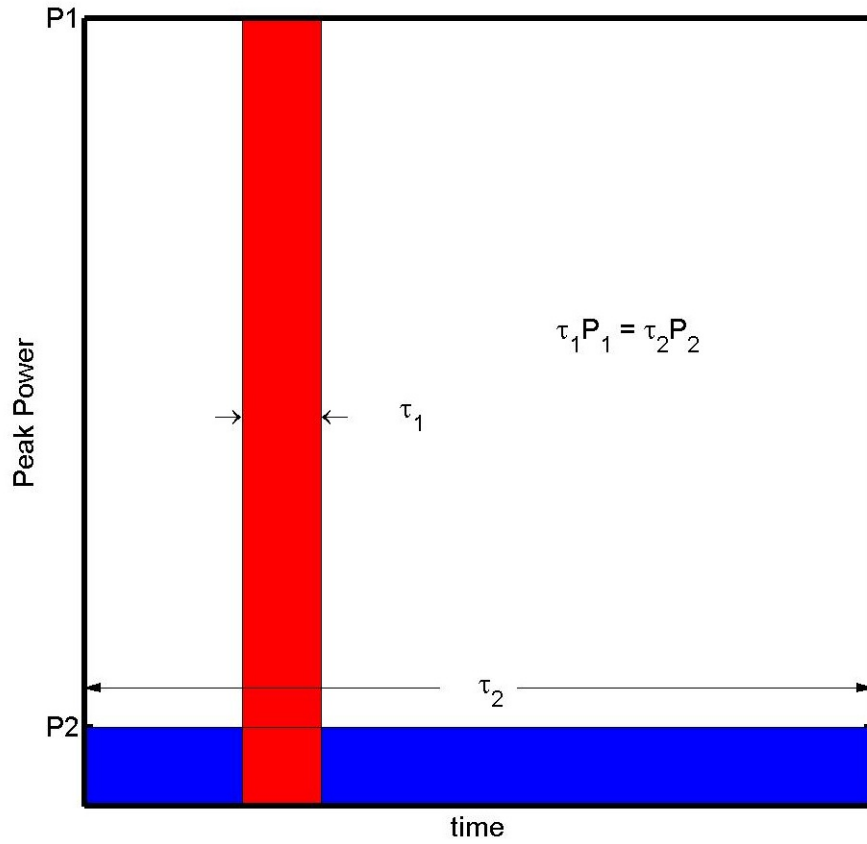


Figure 1.2: Two different waveforms with equal average power

return arrives while the radar is still transmitting. Range resolution is also an important metric that is affected by the length of τ .

Range resolution describes a radar's ability to distinguish between two closely spaced objects. It is the minimum separation between targets that the radar can still resolve distinct targets. The range resolution of an unmodulated pulse (a simple sine wave) with a bandwidth BW is

$$\Delta R = \frac{c}{2BW} \quad (1.3)$$

The bandwidth of a sine wave pulse is determined by τ . Mathematically, a sine wave pulse is modeled in time and frequency by the following Fourier transform pair:

$$\text{rect}\left(\frac{t}{\tau}\right) \cos(\omega_0 t) \longleftrightarrow \frac{1}{2} \left[\tau \text{sinc}\left(\frac{\tau(\omega + \omega_0)}{2\pi}\right) + \tau \text{sinc}\left(\frac{\tau(\omega - \omega_0)}{2\pi}\right) \right] \quad (1.4)$$

The pulse width is modeled by the rectangle function. A rectangle function in time transforms to a sinc function in frequency. So, a simple sine wave pulse transforms to two sinc functions centered at $\pm\omega_0$ in the frequency domain. The bandwidth of this function is approximately inversely proportional to the pulsewidth. This means that we can replace BW in equation 1.3 with $\frac{1}{\tau}$. The range resolution equation for a sine wave pulse can now be rewritten as

$$\Delta R = \frac{c\tau}{2} \quad (1.5)$$

To increase the average power, τ needs to be lengthened. To improve range resolution, τ needs to be shorter. These two diametric properties make using a simple sine wave pulse impractical. To solve the problem, the bandwidth of the pulse needs to be separated from its dependence on the pulsewidth, or the peak power needs to be increased. As previously mentioned, P_t is usually limited by hardware. Fortunately, there is a plethora of modulated waveforms that have a higher bandwidth than a sine wave. The technique of using a pulse with a modulated waveform with a higher bandwidth than an unmodulated pulse is pulse compression.

1.2.2 Pulse Compression

Pulse compression waveforms can be separated into four different categories: linear frequency modulation (LFM), non-linear frequency modulation, time-frequency modulation, and phase-coded waveforms. Each type of modulation has advantages and disadvantages. Radar requirements determine the type of waveform that is selected. The radar system

analyzed in these pages implements LFM; therefore, the description of pulse compression waveforms is limited to LFM.

LFM means that the frequency is modulated linearly with respect to time. Frequency can increase or decrease linearly. LFM is also commonly referred to as chirp. This comes from the chirping sound that the waveform makes in the audio range. LFM is defined as

$$x(t) = A \operatorname{rect}\left(\frac{t}{\tau}\right) \cos(2\pi f_0 t + \pi\alpha t^2) \quad (1.6)$$

where τ is the pulsewidth, f_0 is the carrier frequency, and α is the constant slope of the LFM. α can be positive or negative. From this equation, the phase modulation term is

$$\phi(t) = \pi\alpha t^2 \quad (1.7)$$

The frequency modulation is the derivative of the phase modulation term with respect to time. It is

$$f(t) = \frac{1}{2\pi} \frac{d\phi}{dt} = \alpha t \quad (1.8)$$

Equation 1.8 shows that the frequency does indeed change linearly with respect to time. A plot that allows an easy way to visualize an LFM waveform is a frequency vs. time plot. Figure 1.3 is simply a straight line. This type of plot is important to introduce because it is used extensively in explaining how to process LFM returns.

Deriving the Fourier transform of an LFM is not straightforward. A useful characterization of LFM is the time-bandwidth product. The time-bandwidth product is the product of the pulsewidth and the frequency range swept during the pulse. For time-bandwidth products much greater than one, the spectrum of the LFM approaches a rectangle function,

$$|X(f)| = \operatorname{rect}\left(\frac{f}{BW}\right) \quad \text{for } \tau BW \gg 1 \quad (1.9)$$

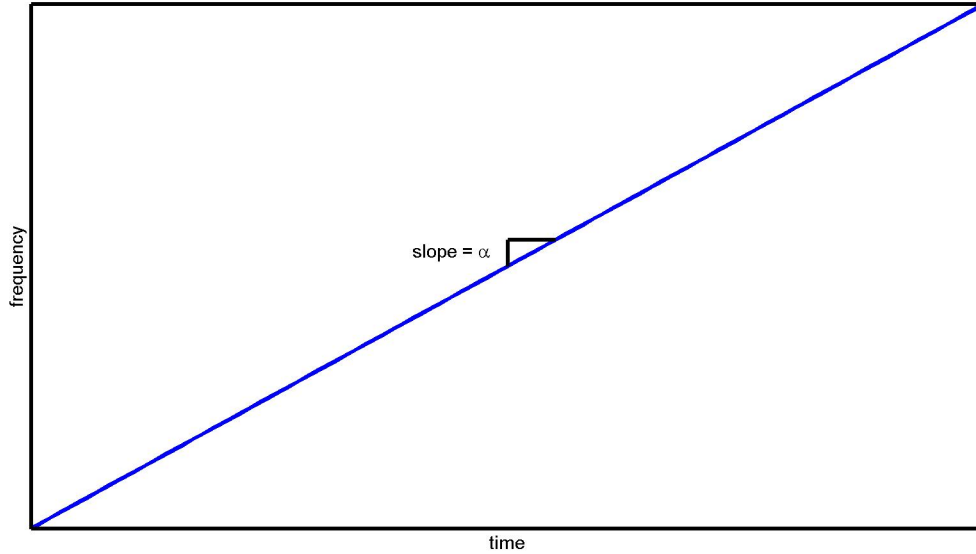


Figure 1.3: Frequency vs. time of an LFM waveform

If the time-bandwidth product is not high enough, there is amplitude variation in the frequency spectrum. An example of an LFM waveform with a time-bandwidth product greater than 1000 in the time domain and frequency domain is given in Figure 1.4. The variation is negligible and is barely conceived at 0 Hz and the corners of the rectangle.

1.2.3 Stretch Processing

Stretch processing is an alternative to using a matched filter. The analysis and implementation of stretch processing is expounded in depth in later sections. Here, an introduction is given. “Stretch pulse compression is a technique for performing LFM pulse compression of wideband waveforms using a signal processor with bandwidth that is much smaller than the waveform bandwidth, without loss of SNR or range resolution [1].” Whereas a matched filter response is a spike in time corresponding to a target range, stretch processing produces a single-tone frequency corresponding to a target range. Alternatively, stretch processing can give another LFM waveform with a much smaller bandwidth. This is useful to reduce the bandwidth of the LFM waveform to a bandwidth that can be sampled by an ADC. It is

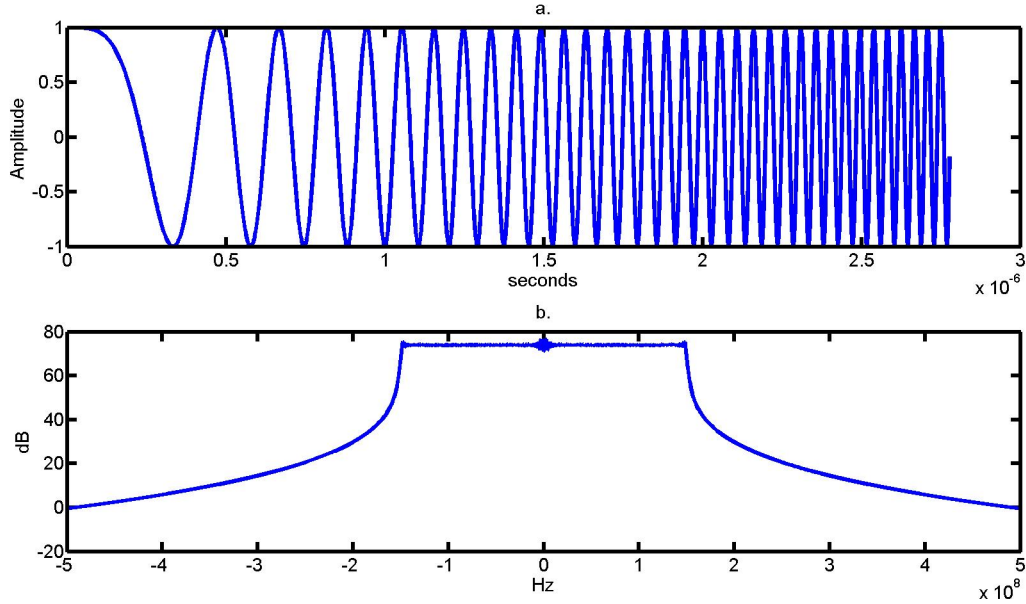


Figure 1.4: a) Time domain of LFM, b) Frequency domain of LFM

then further processed in the digital domain. Stretch processing multiplies or mixes a return LFM with a reference LFM. The process can be implemented in the analog domain with a mixer that can support the bandwidth of the LFM waveforms. To understand how this works, recall the following two mathematical relations:

$$\cos(a) \cdot \cos(b) = \frac{1}{2} [\cos(a + b) + \cos(a - b)] \quad (1.10)$$

and time shifting a function in time

$$f(t) \text{ delayed by } \tau \text{ seconds} = f(t - \tau) \quad (1.11)$$

Multiplying two cosine wave together results in the sum and difference of the frequencies. When two LFM waveforms with the same α are multiplied together the difference is constant. The single-tone result is dependent only on the relative time difference of the two waveforms. In stretch processing, an LFM waveform is transmitted. After waiting for a set amount of

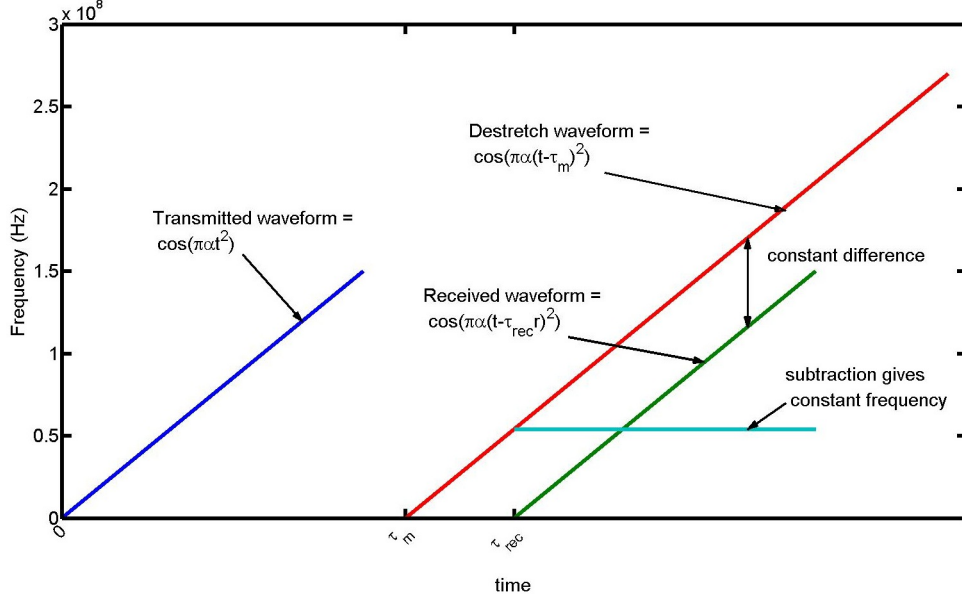


Figure 1.5: Stretch Processing

time after transmission, a reference or destretch waveform starts at one input to a mixer. The destretch waveform is mixed with any received waveforms. The result is a single-tone frequency that is proportional to the time difference of the two waveforms. Figure 1.5 illustrates the process using a frequency vs. time plot. Note that the destretch waveform needs to be much longer than the transmitted waveform.

Mathematically, stretch processing is modeled by

$$\begin{aligned}
 & 2 \cos(\alpha\pi(t - \tau_R)^2) \cdot \cos(\alpha\pi(t - \tau_M)^2) \\
 & = \cos(\alpha\pi(t - \tau_R)^2 + \alpha\pi(t - \tau_M)^2) + \cos(\alpha\pi(t - \tau_R)^2 - \alpha\pi(t - \tau_M)^2) \quad (1.12)
 \end{aligned}$$

where τ_R is the time delay of the received waveform and τ_M is the time delay of the reference waveform. The time delays are relative to when the waveform was transmitted. In equation 1.12, the sum of the two terms can be ignored. In the frequency domain, it is “stretched” enough that its power is insignificant. The subtraction term is the desired result. When the

two quadratic terms are expanded and combined, the result is

$$\cos(2\pi\alpha(\tau_R - \tau_m)t + \theta) \quad (1.13)$$

where θ is all of the non-time-dependent terms combined into a constant phase term. The frequency of equation 1.13 is

$$f_M = \alpha(\tau_R - \tau_M) \quad (1.14)$$

The target range is calculated from this frequency. The only unknown in the equation is τ_R . τ_M , the reference waveform delay, represents the delay from a known relative range.

$$\begin{aligned} \Delta\tau &= \tau_R - \tau_M \\ &= \frac{f_M}{\alpha} \end{aligned} \quad (1.15)$$

The equation for the relative range is

$$\begin{aligned} \Delta r &= (R - R_0) \\ &= \frac{c}{2}\Delta\tau \end{aligned} \quad (1.16)$$

$$\text{where } R_0 = \frac{c}{2}\tau_M \quad (1.17)$$

The range window of the radar is limited by the pulsewidth of the destretch waveform. The pulsewidth is often limited by either the highest frequency that can be generated by the waveform generator or the bandwidth that can be handled by the stretch processing. Also, the Doppler frequency is coupled to the stretch frequency. It cannot be separated and is an error. The Doppler frequency can be determined and compensated for by processing multiple target returns in the digital domain.

Chapter 2

System

2.1 Simulation

To ensure that the ROC works as expected when designed, a high-level simulation was designed. The main purpose is to model the signal development in the transmitter and the receiver. Once designed, the simulation can quickly simulate different values for the carrier frequency, IF frequency, LFM bandwidth, LFM slope, and stretch processing implementation. Filter specifications can also be determined from the simulation. Figure 2.1 is a block diagram of the ROC. The bottom half of the diagram is the transmitter. The encircled numbers mark important points in the signal analysis. For the following equations and figures, the amplitude is neglected. Correct amplification is assumed. The waveform shape and frequency content along the transmitter and receiver is the focus. Also, to make the math easier to follow, the waveform windows or rectangle functions are ignored in the equations. These are very important in much of the radar analysis, but not much is lost by leaving them out in the equations to go along with the MATLAB simulation.

2.1.1 Transmitter

Understanding the concept of quadrature mixing is important to understanding the radar operation. Quadrature mixing requires two mixers with the same local oscillator (LO) frequency, but one LO signal has a phase offset of 90° . In the radar diagram, there are three different occurrences of quadrature mixing. They appear as parallel mixers in the diagram. Notice that each quadrature pair has an **I** (inphase) signal at one LO port and a **Q** (quadrature phase) at the other LO port. Recall that $\cos(a) = \sin(a + 90^\circ)$; therefore, the **I** signal can be considered a cosine, and the **Q** signal is a sine. Quadrature mixing allows

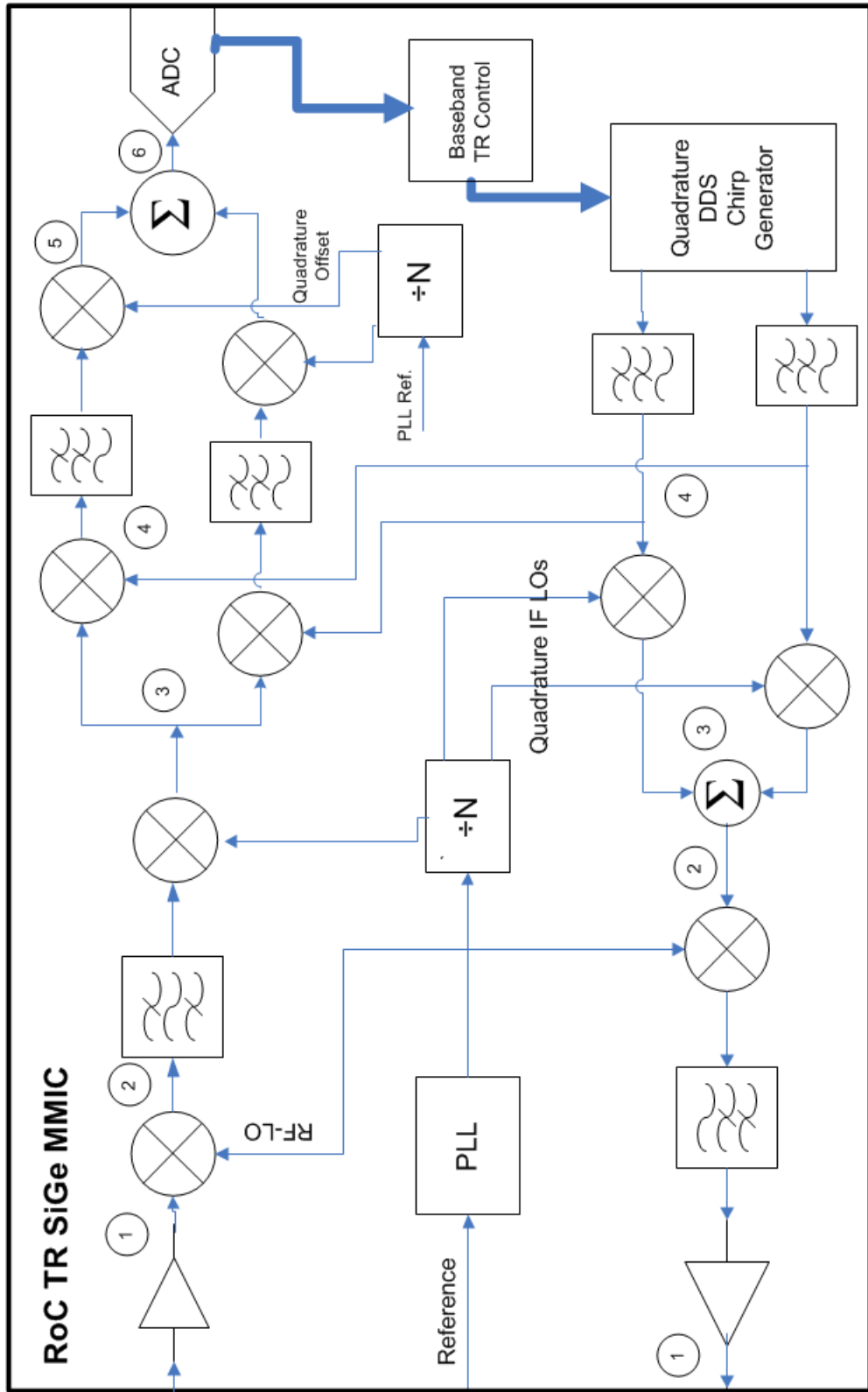


Figure 2.1: ROC block diagram

canceling either the sum or difference output of a mixer by either adding or subtracting the **I** and **Q** mixing results.

$$[\cos(a) \cdot \cos(b)] \pm [\sin(a) \cdot \sin(b)] = \frac{1}{2}[\cos(a+b) + \cos(a-b)] \pm \frac{1}{2}[\cos(a-b) - \cos(a+b)] \quad (2.1)$$

Quadrature mixing is especially useful for canceling a mixing result that is too close to the desired frequency to be filtered.

Node 4

The LFM waveform generation begins with the DDS in the lower right corner Figure 2.1. The DDS has two outputs— an in-phase and quadrature phase LFM waveform. These waveforms need to be filtered with a low pass filter (LPF) to remove spurious frequencies inherent in the DDS output. Refer to Figure 1.4 for the visual representation of the baseband LFM waveform.

I output:

$$\cos(\alpha\pi t^2) \quad (2.2)$$

Q output:

$$\sin(\alpha\pi t^2) \quad (2.3)$$

Node 3

The next block in the signal path is a quadrature mixing stage that mixes the baseband LFM waveform with the IF. The quadrature mixing is used to cancel the lower sideband of the LFM waveform in the next step. The only difference in the outputs of the two waveforms is the phase of one of the terms. Figure 2.2 is the frequency spectrum of the output of one of the mixers.

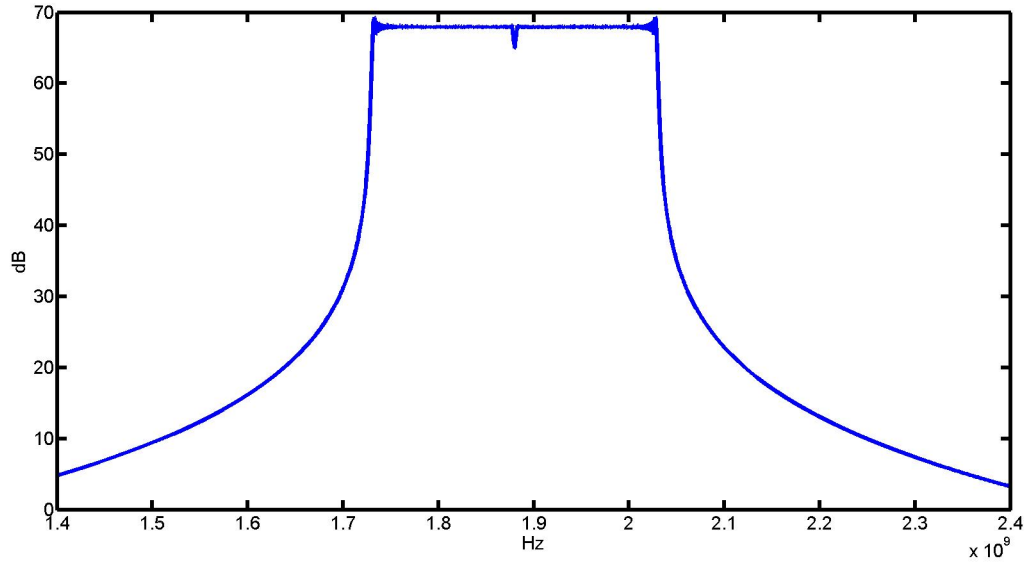


Figure 2.2: LFM waveform at IF

The output of the **I** mixer:

$$\cos(\alpha\pi t^2) \cdot 2 \cos(\omega_{IF}t) = \cos(\omega_{IF}t + \alpha\pi t^2) + \cos(\omega_{IF}t - \alpha\pi t^2) \quad (2.4)$$

The output of the **Q** mixer:

$$\sin(\alpha\pi t^2) \cdot 2 \sin(\omega_{IF}t) = \cos(\omega_{IF}t - \alpha\pi t^2) - \cos(\omega_{IF}t + \alpha\pi t^2) \quad (2.5)$$

Node 2

After the quadrature mixing, the **I** and **Q** channels are combined with a summer. This cancels the lower sideband. Before the summation, the signal has two chirps. The chirps start at the IF, and one increases in frequency while the other decreases in frequency at the same slope. This waveform could be used for transmission. This system removes the lower sideband to reduce the power and bandwidth of the system. Figure 2.3 is the signal spectrum

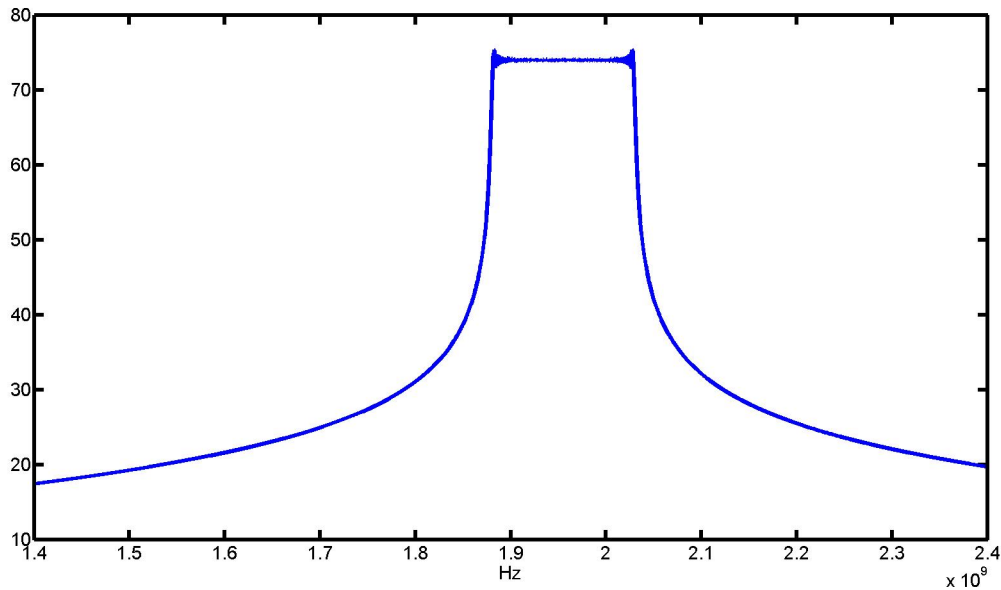


Figure 2.3: Single-sideband LFM waveform at IF

after the cancellation. It is exactly half the bandwidth of the bandwidth in Figure 2.2.

$$\cos(\omega_{IF}t + \alpha\pi t^2) + \cos(\omega_{IF}t - \alpha\pi t^2) - \cos(\omega_{IF}t - \alpha\pi t^2) + \cos(\omega_{IF}t + \alpha\pi t^2) = \cos(\omega_{IF}t + \alpha\pi t^2) \quad (2.6)$$

Node 1

Once the signal is single-side band at IF, it is amplified with a variable gain amplifier. The signal is now ready to be up-converted to RF. The signal is mixed with the RF from the on-chip phase-lock loop (PLL). After mixing, the signal is filtered to get rid of the unwanted RF-IF frequency component. The last on-chip block that the signal goes through is the power amplifier driver. It prepares the signal to be input into an external power amplifier

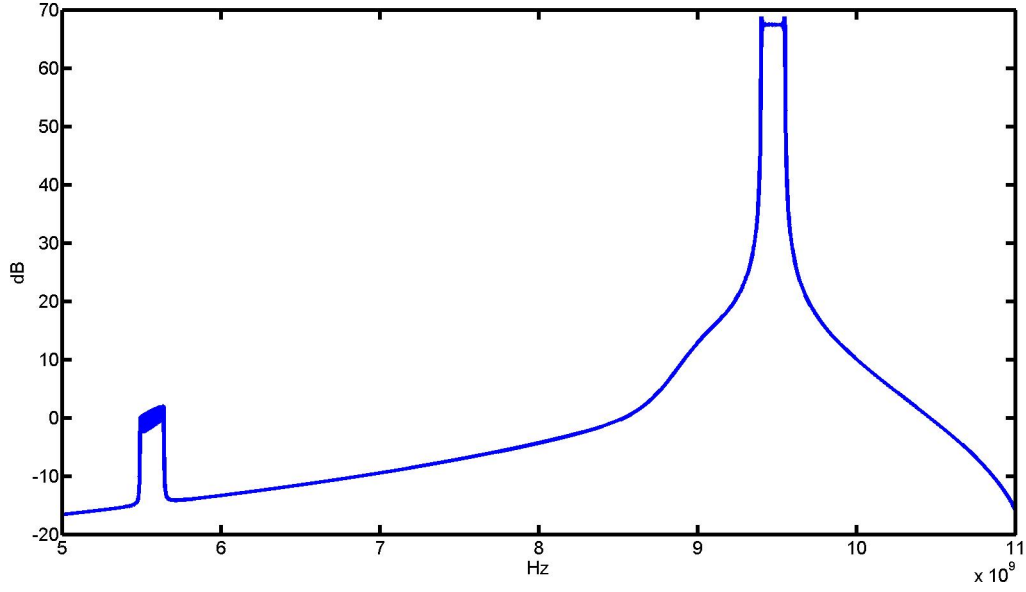


Figure 2.4: Filtered RF waveform in the frequency domain

before transmission. The final on-chip signal shown in Figure 2.4 is

$$\begin{aligned}
 \cos(\omega_{IF}t + \alpha\pi t^2) \cdot 2 \cos(\omega_{LO}t) &= \cos(\omega_C t + \alpha\pi t^2) + \overbrace{\cos((\omega_{LO} - \omega_{IF})t - \alpha\pi t^2)}^{\text{Filtered}} \\
 &= \cos(\omega_C t + \alpha\pi t^2)
 \end{aligned} \tag{2.7}$$

2.1.2 Receiver

Node 1

The received pulse is similar to the transmitted pulse. It has a time delay τ_R , a Doppler frequency ω_D , phase noise ϕ , and noise v_N added to the transmitted pulse. Equation 2.8 is the received signal at node 1:

$$s_{rec} = \text{rect}\left(\frac{t - \tau_R}{T}\right) \cos((\omega_C + \omega_{dop})(t - \tau_R) + \alpha\pi(t - \tau_R)^2 + \phi) + v_N \tag{2.8}$$

The effects of the noise, phase noise, Doppler frequency, and rectangle function will be ignored for the rest of the simulation results. For now, just the signal processing is shown. Each of these components will be examined in-depth in Chapter 3. After removing these components the signal looks like

$$s_{rec} = \cos(\omega_C(t - \tau_R) + \alpha\pi(t - \tau_R)^2) \quad (2.9)$$

Node 2

After the received signal is amplified by either the wideband or narrowband LNA, it is mixed down to the IF frequency and filtered. The filter requirements were too stringent to be efficiently implemented on-chip. This IF SAW filter is one of the few components in the system that is not integrated. The bandwidth of this filter is as close to the bandwidth of the chirp as possible. The output of a received waveform at the SAW filter should be the same as Figure 2.3. The equation for the signal after the filter is

$$s_{IF} = \cos(\omega_{IF}(t - \tau_R) + \alpha\pi(t - \tau_R)^2) \quad (2.10)$$

Node 3

At node 3 the signal is down-converted from IF to baseband. This means that the signal is mixed with the IF. The $2\omega_{IF}$ term is filtered out later by a baseband filter. The base band signal is

$$\begin{aligned} s_{BB} &= \cos(\omega_{IF}(t - \tau_R) + \alpha\pi(t - \tau_R)^2) \cdot 2 \cos(\omega_{IF}t) \\ &= \cos(2\omega_{IF}(t - \tau_R) + \alpha\pi(t - \tau_R)^2) + \cos(\alpha\pi(t - \tau_R)^2) \end{aligned} \quad (2.11)$$

Figure 2.5 shows the spectrum of the recieved baseband waveform.

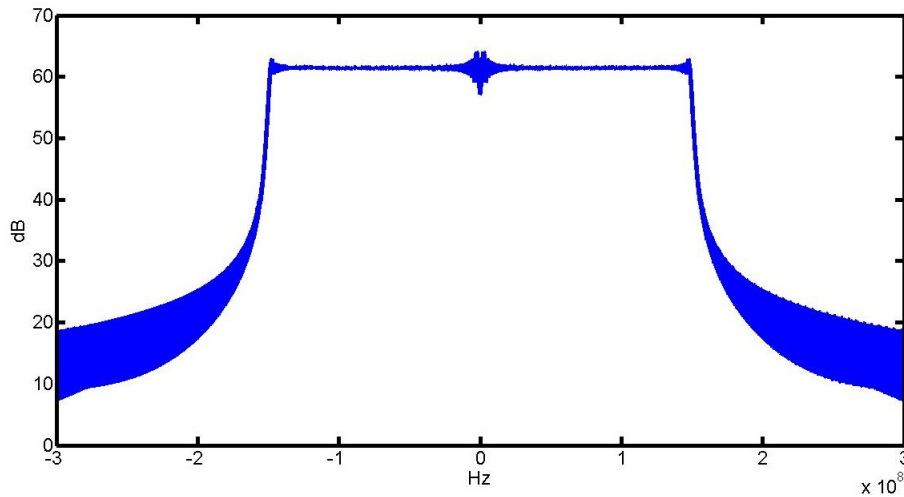


Figure 2.5: Received BB waveform in the frequency domain

Node 4

After the signal is down-converted to baseband, it is amplified with a variable gain baseband amplifier. Next, the stretch processing occurs. To understand why this particular implementation of stretch processing, two conditions need to be described. Firstly, only one ADC is used on the chip. This is for space and power savings. Many systems use two ADCs to convert both the **I** and **Q** channels to the digital domain. Having both **I** and **Q** channels allows for phase information to be retained.

Secondly, the scenario where a return arrives at the ROC before the destretch process begins. For example, a $10\mu s$ chirp arrives at the receiver $1\mu s$ before the destretching begins. $1\mu s$ is lost, but the remaining $9\mu s$ are destretched. This results in a negative frequency corresponding to a negative range. Some power is lost in returns coming in early, but they may still have significant power. Figure 2.6 demonstrates returns coming in $\pm 3\mu s$ from the destretch waveform. The results are frequencies at $\pm 27MHz$.

The concept of a negative range or negative frequency may seem odd at first glance. Think of the destretch waveform as a reference range. For example, if the destretch waveform starts $30\mu s$ after transmission, it represents a range of $\frac{c \cdot 30\mu s}{2} = 4.5km$. The frequencies from

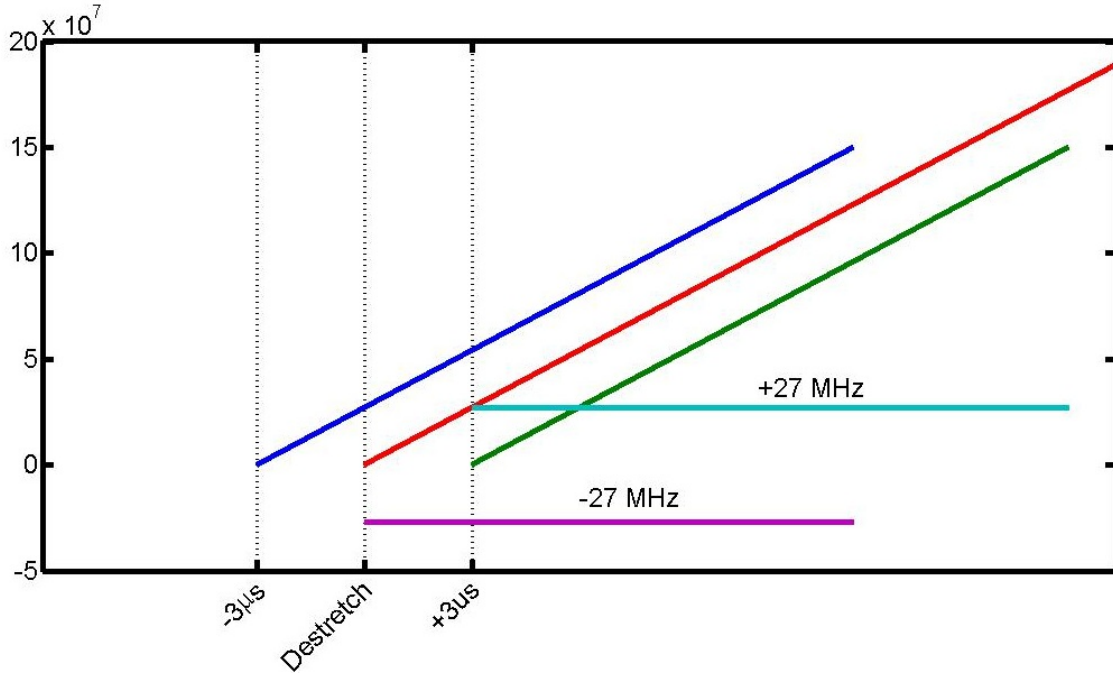


Figure 2.6: Destretching with a return at $\pm 3\mu s$

the destretching represent ranges relative to the relative range of the destretch waveform. In this example, any range less than 4.5 km will show up as a negative frequency. Ideally, the distinguishing of positive and negative ranges is done in the digital domain, but this requires **I** and **Q** channels that are not available for this system. With a single ADC, positive and negative frequencies are practically the same. This range ambiguity has to be solved in the analog domain.

Quadrature destretch waveforms are used for the stretch processing. Eventually, this allows for the canceling of a sideband that allows the distinguishing of negative and positive frequencies. At node 4, the output of the two mixers are as follows.

I Mixer:

$$\begin{aligned}
 s_I &= 2 \cos(\alpha\pi(t - \tau_M)^2) \cdot \cos(\alpha\pi(t - \tau_R)^2) \\
 &= \cos(2\pi\alpha(\tau_R - \tau_M)t - \theta) + \cos(\alpha\pi(t - \tau_R)^2 + \alpha\pi(t - \tau_M)^2)
 \end{aligned} \tag{2.12}$$

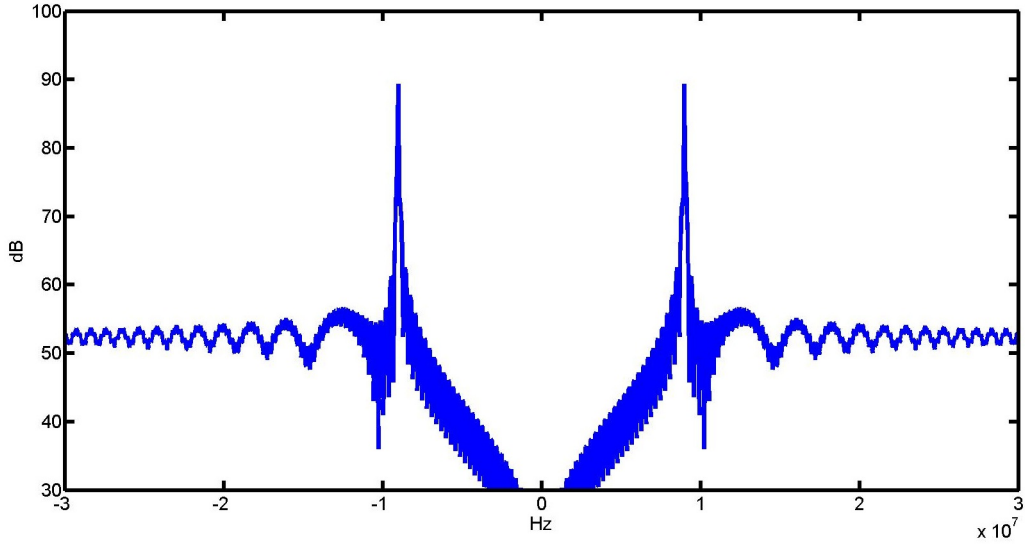


Figure 2.7: Example of a received waveform after destretching

Q Mixer:

$$\begin{aligned}
 s_I &= 2 \sin(\alpha\pi(t - \tau_M)^2) \cdot \cos(\alpha\pi(t - \tau_R)^2) \\
 &= \sin(2\pi\alpha(\tau_R - \tau_M)t - \theta) + \sin(\alpha\pi(t - \tau_R)^2 + \alpha\pi(t - \tau_M)^2)
 \end{aligned} \tag{2.13}$$

τ_M is the time delay of the destretch signal and θ is a phase term introduced by the stretch processing. The difference term from the two mixers is the single-tone frequency that is wanted. The sum term overlaps the single-tone frequency. This adds to the noise, but its impact is nearly negligible because its power is so spread out. Figure 2.7 is an example of a received waveform after stretch processing. Note that it is impossible to know if the relative range is negative or positive.

Node 5

Both channels have an LPF after the stretch processing. These filters set the range of frequencies that the system can process. For example, a filter with a cutoff of 10 MHz means

that only ranges correspond to a frequency range of ± 10 MHz are acceptable. The purpose of the filters will become more clear.

After the filters, each channel is mixed with an offset frequency. The offset frequency needs to be at least the frequency of the cutoff of the baseband filter. This ensures that any negative frequency is now positive. For example, a frequency of -9 MHz becomes a frequency of 1 MHz. This is why the filters are needed. Any frequency that is still negative after mixing will give a false range reading. A frequency of -11 MHz is mixed with 10 MHz to get to -1 MHz. A single ADC will sample this frequency as 1 MHz– the same as 1 MHz in the previous example that resulted from mixing up -9 MHz. The figures demonstrating this part of the system use a filter with a cutoff of 10 MHz and an offset frequency of 10 MHz. To keep the math easy to follow, the sum terms are dropped from the equations 2.14 and 2.15. The results of the mixing are as follows:

I Mixer:

$$\begin{aligned} s_I &= \cos(2\pi\alpha(\tau_R - \tau_M)t - \theta) \cdot 2 \cos(\omega_{offset}t) \\ &= \cos(\omega_{offset}t + 2\pi\alpha(\tau_R - \tau_M)t - \theta) + \cos(\omega_{offset}t - 2\pi\alpha(\tau_R - \tau_M)t + \theta) \end{aligned} \quad (2.14)$$

Q Mixer:

$$\begin{aligned} s_Q &= \sin(2\pi\alpha(\tau_R - \tau_M)t - \theta) \cdot 2 \sin(\omega_{offset}t) \\ &= -\cos(\omega_{offset}t + 2\pi\alpha(\tau_R - \tau_M)t - \theta) + \cos(\omega_{offset}t - 2\pi\alpha(\tau_R - \tau_M)t + \theta) \end{aligned} \quad (2.15)$$

Now any negative frequencies have been mixed to a positive frequency. After mixing with the offset frequency, there is still an unwanted frequency term– the difference of the offset and the stretch processing frequency. Figure 2.8 is Figure 2.7 mixed with the offset frequency.

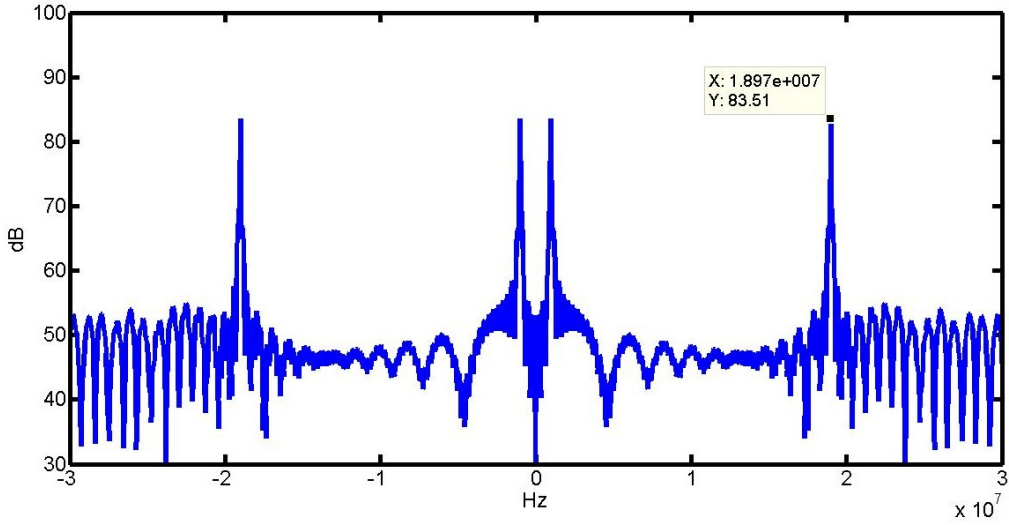


Figure 2.8: Destretched waveform return mixed with the offset frequency

Node 6

Due to the quadrature stretch processing and offset mixing, the unwanted frequency can be canceled by subtracting Equation 2.14 and Equation 2.15. Figure 2.9 is the result of subtracting the two channels. This is the final frequency output that is sampled by the ADC where $\Delta\tau = \tau_R - \tau_M$.

$$\begin{aligned}
 s &= s_I - s_Q \\
 s &= \cos(\omega_{offset}t + 2\pi\alpha(\Delta\tau t - \theta))
 \end{aligned}
 \tag{2.16}$$

2.2 Cadence Simulation

The ROC was designed using Cadence IC 6.1. The system parts were designed by multiple designers. Each system part was tested individually by its designer. Testing the system as a whole is still very important though. As with any complex system, testing is also complex and non-trivial. Some system parts cannot be easily included in a full system test. For example, the PLL and the ADC are better tested separately. The important

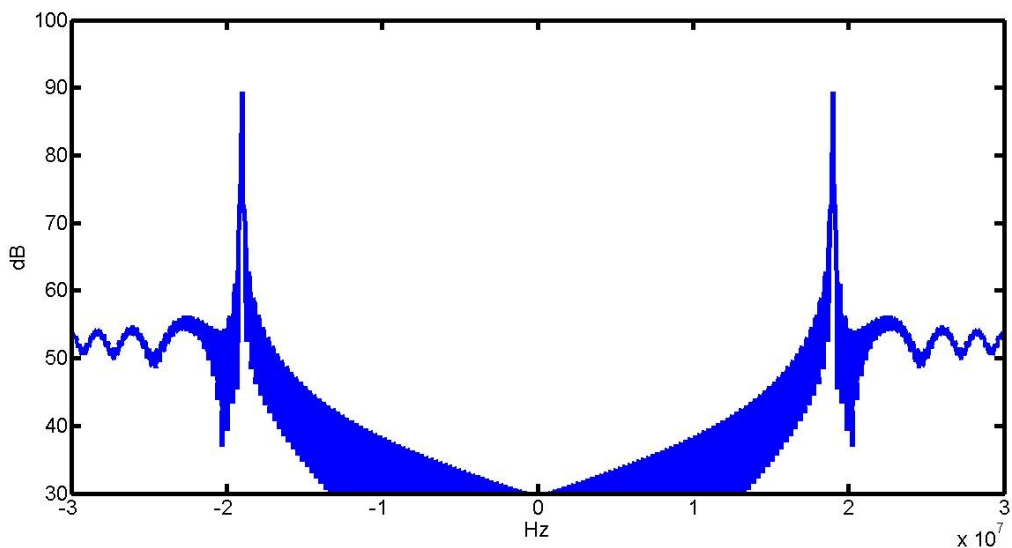


Figure 2.9: Result of subtracting the two channels

consideration for any part left out of the system simulation is the interconnect between it and the rest of the system.

2.2.1 TX Simulation

Ideally, the a full length transient simulation of a chirp could be performed on the system from the DAC at the output of the DDS to the power amplifier driver. This type of simulation is not practical though. The problem is that changing time domains from baseband to X-band make simulating a significant length of time at baseband prohibitive. The time step for simulating a waveform at X-band is very small relative to the length of time required to simulate a baseband waveform. Instead of simulating a chirp, a series of single tone frequencies that span the chirp bandwidth is simulated.

To test the basic operation of the transmitter, baseband frequencies of 45 MHz, 120 MHz and 195 MHz were simulated with a transient simulation for 130 ns. The IF is 1.76 GHz and the PLL is $1.76 \text{ GHz} \cdot 4 = 7.04 \text{ GHz}$. All three tests were successful. The amplitude of the output frequencies are similar and all spurs are about 40 dB below the output. Also there are no spurs within the LFM bandwidth. The output spectrum has a frequency resolution

of 10 MHz; therefore, the 45 MHz and 195 MHz simulations have the energy split between 40 MHz and 50 MHz, and 190 MHz and 200 MHz respectively. Figure 2.10, Figure 2.11, Figure 2.12 give the results of the simulation.

2.2.2 RX Simulation

Unlike the transmitter, the receiver needs to be tested with an LFM waveform. It does not need to be tested from RF to the ADC with an LFM waveform though. This creates the aforementioned problem of simulation resources. The important part of the receiver that needs to be tested is the stretch processing. This spans from IF to the ADC input. A transient simulation of a full chirp from IF to baseband requires a lot of resources. A six core computer with 8 GB of ram requires several days for each simulation. Since each simulations requires so many resources, multiple received chirps were simulated at once.

As input to the simulation, a series of chirps spanning the whole width of the 20 MHz range was generated with a MATLAB script. There are several things to watch for in the simulation results. The processing gain of the stretch processing needs to be close to what is expected. The destretched frequencies need to be narrowband. The image frequencies that may appear because of quadrature stretch processing need to be well below the destretched frequencies. Figure 2.13 is the received LFM waveforms at IF. The individual echoes are not distinguishable at this point.

The simulation was performed twice. Once with low gain and again with high gain in the baseband VGA. The results are in Figure 2.14 and Figure 2.15. In the high gain results, image frequencies are starting to emerge in the spectrum. They are still more than 20 dB below the desired frequency though. Table2.1 shows the delays and destretched frequency values of the simulation.

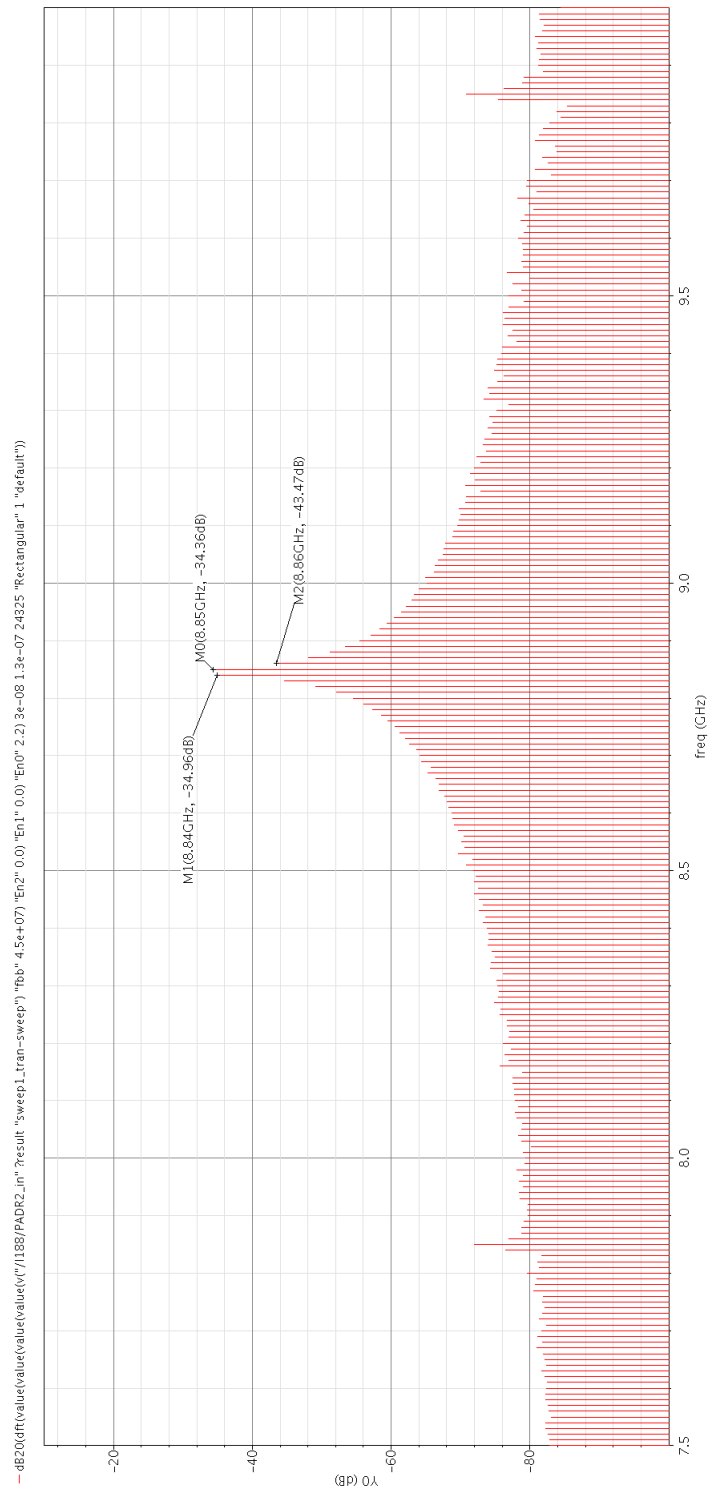


Figure 2.10: 45 MHz transmitter simulation result

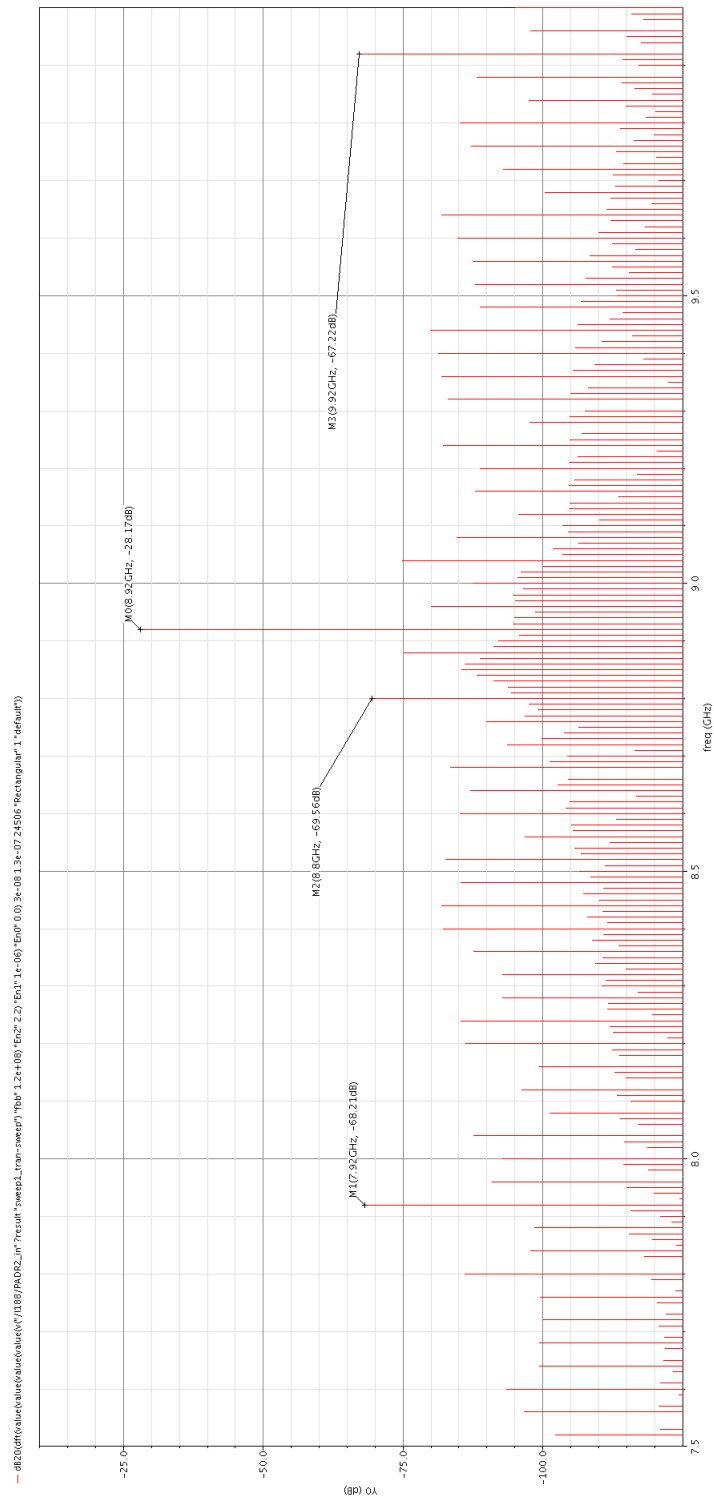


Figure 2.11: 120 MHz transmitter simulation result

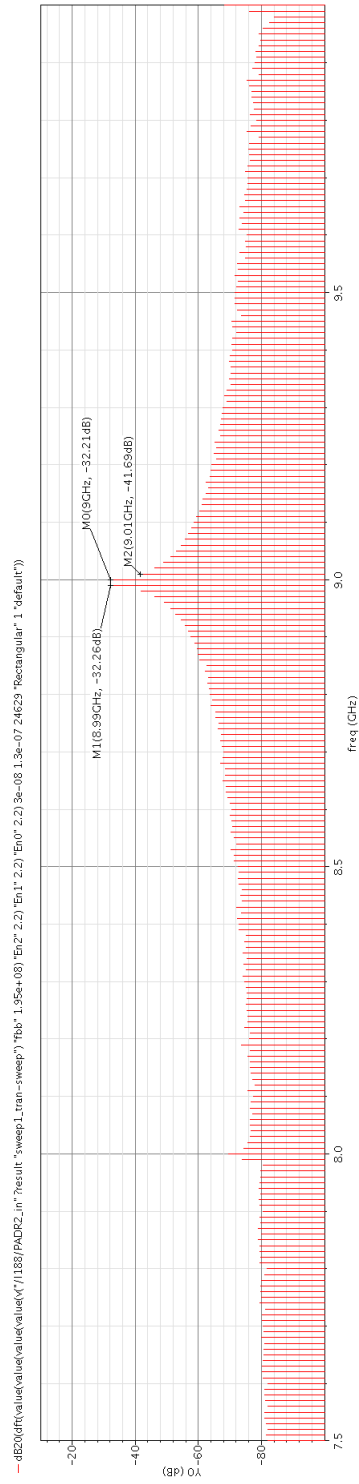


Figure 2.12: 195 MHz transmitter simulation result

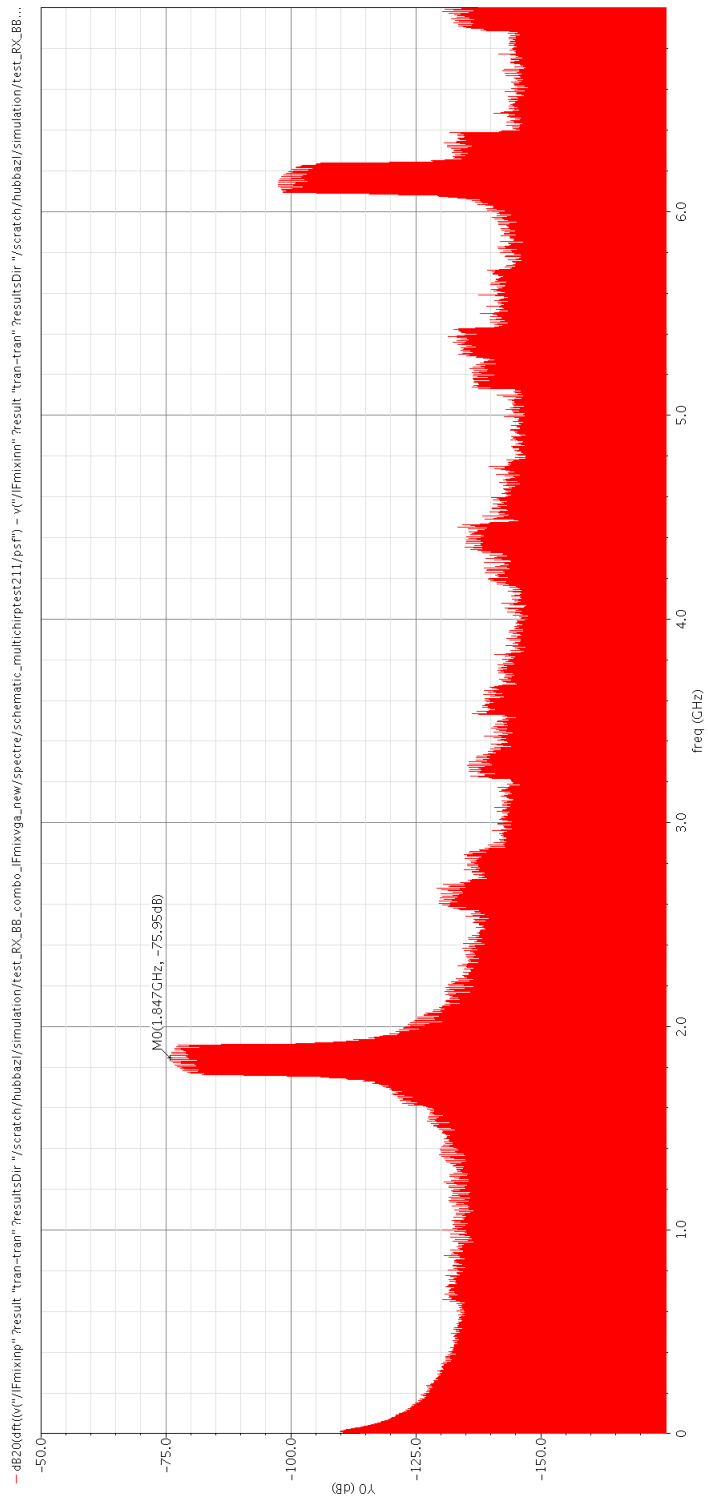


Figure 2.13: RX IF VGA output

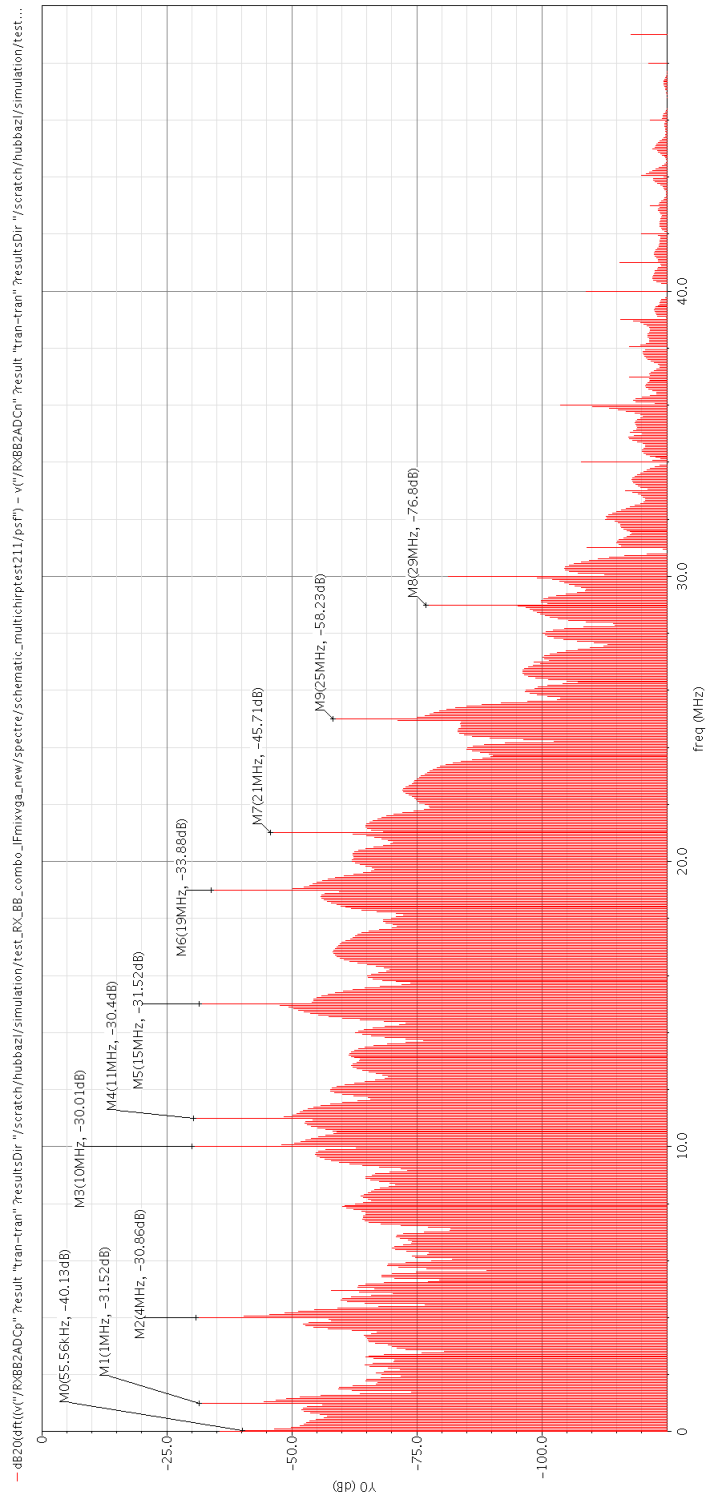


Figure 2.14: RX simulation results for low gain

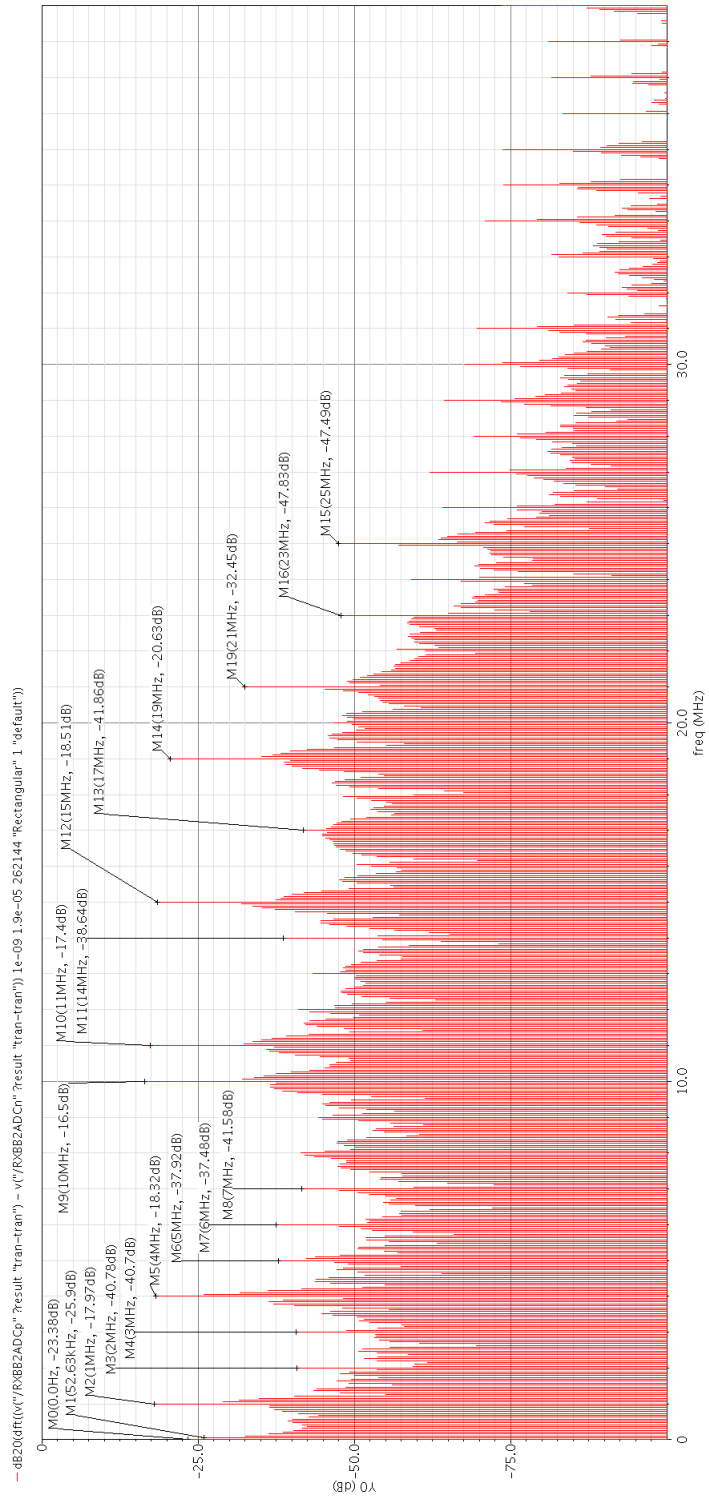


Figure 2.15: RX simulation results for low gain

Table 2.1: Delay and frequency values for RX simulation

Delay (s)	Frequency (Hz)
-1.66E-06	-4.94E+06
-1.11E-06	1.00E+03
-1.00E-06	1.00E+06
-6.67E-07	4.00E+06
0.00E+00	1.00E+07
1.11E-07	1.10E+07
5.56E-07	1.50E+07
1.00E-06	1.90E+07
1.66E-06	2.49E+07

Chapter 3

Stretch Processing

3.1 Pulsewidth and LFM Start Frequency

There are many factors that determine the performance and characteristics of the ROC. The maximum system bandwidth, maximum slope of the LFM, the baseband filter, and the performance of the ADC can all be limiting factors for the radar performance. For example, if the LFM bandwidth is 150 MHz, each component in the transmitter and the receiver until the stretch processing is complete must be able to handle this bandwidth without distortion. The DDS limits the slope of the LFM waveform; therefore, it sets the minimum pulsewidth required for a given bandwidth.

Discovering the limiting factors of the system and optimizing system performance is very important. Remember that range resolution and pulse length can be set independently. If pulse length and bandwidth are defined by system specifications such as average power and range resolution, simply divide the bandwidth by the pulsewidth to calculate the slope of the LFM waveform. Make sure that the time-bandwidth product is much greater than one to avoid amplitude ripple in the frequency domain.

3.1.1 Pulsewidth

Setting the delay τ_M and the length of the destretch waveform is required. If the pulsewidth is too long, power and time are wasted. There is no need to process returns that will promptly be filtered by the baseband filter. If the pulsewidth is too short, power is lost in the stretch processing. For the way stretch processing is implemented on the ROC, the destretch waveform needs a different start frequency to make sure that all of the power is captured from returns in the range window. The minimum delay of τ_M is determined

by the length of the transmitted pulse. The ROC cannot start receiving until it is done transmitting.

The acceptable frequency range for the ROC is $\pm f_{cutoff}$ of the baseband filter. The optimum pulsewidth to cover this frequency range is

$$T = \text{time of transmitted pulse} + \frac{2f_{cutoff}}{\alpha} \quad (3.1)$$

3.1.2 Offset Frequency

Equation 3.1 only holds true for the way stretch processing is implemented on the ROC. To use the minimum pulsewidths and make sure all returns in the acceptable range are fully processed a frequency offset between the two pulses is required. The offset is equal to the maximum possible frequency value.

$$f_{start} = f_M^{max} \quad (3.2)$$

The offset frequency has to be taken into account when calculating the relative range.

$$\Delta r = \frac{c}{2} \frac{(f_M - f_{start})}{\alpha} \quad (3.3)$$

Without any offset frequency, a negative relative range implies that the return arrives at the receiver before the destretch waveform starts. The offset makes sure that the earliest processed return occurs at the beginning of the stretch processing. Examine Figure 3.1 to see why power is wasted without an offset frequency.

3.1.3 Range Window

One comparative disadvantage of stretch processing compared to matched filtering is the small range window. Several factors affect the size of the range window such as α and

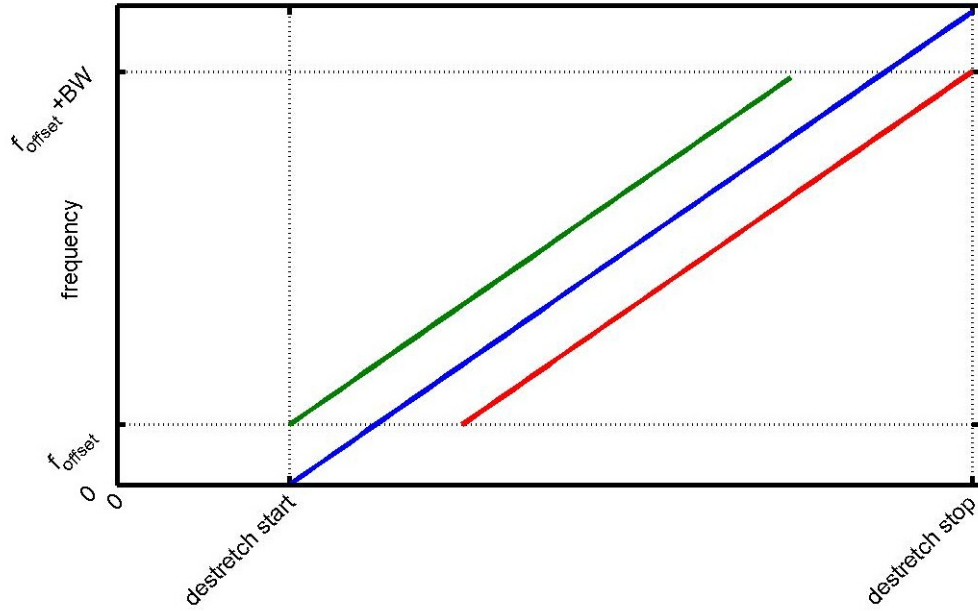


Figure 3.1: LFM with frequency offset

the bandwidth of the baseband filter. The equation for the range window is

$$\text{range window} = \frac{c}{2} \frac{f_{range}}{\alpha} \quad (3.4)$$

where f_{range} is the total bandwidth of frequencies corresponding to a stretch processed frequency. For the ROC, $f_{range} = 2f_{cutoff}$. α can be varied by the DDS. When α is varied, the pulsewidth of the transmitted waveform also has to be varied to maintain a constant range resolution. If the frequency is increasing slower (smaller slope), more time is needed to cover the whole bandwidth of the LFM waveform. If the slope is decreased by a factor of 10, the range window and pulsewidth are increased by a factor of 10.

If the range resolution is not set, the pulsewidths can also be varied with the changing slope to achieve a higher PRF.

Table 3.1: Varying α to adjust the range window with constant range resolution

α ($\frac{Hz}{s}$)	Pulsewidth (μs)	Destretch Pulsewidth (μs)	Range Window (m)	Minimum PRF (kHz)
9.00E+12	16.7	18.9	3.33E+02	28.1
9.00E+11	167	189	3.33E+03	2.81
4.50E+12	33.3	37.8	6.67E+02	14.1
4.50E+11	333	378	6.67E+03	1.41

Bandwidth of TX pulse = 150 MHz
 Bandwidth of destretched frequencies = 20 MHz

3.2 Doppler

The Doppler frequency is inseparable from the destretched frequency; therefore, it adds directly to the error. Fortunately this error is generally negligible. A Doppler shift occurs when there is a relative radial velocity between the radar and the target. When the radiating source is moving towards the point of observation, each wave crest (or any other periodic point in the wave) is emitted closer to the point of observation. This causes an increase in the frequency. If the radiating source is moving away from the observation point, the distance between wave crests is increased, causing a decrease in frequency. Detecting Doppler shifts is very important for many radar applications. For the scenario of a moving target, consider the receiver as the point of observation and the target as the source. The target is the source because it re-radiates the transmitted waveform. The Doppler frequency shift can be accurately approximated as

$$f_{dop} = \frac{2v_r}{\lambda} \quad (3.5)$$

When the Doppler frequency is added to the final stretch processed waveform in Equation 2.16, it becomes

$$\cos((\omega_{offset} + \omega_{dop})t + 2\pi\alpha\Delta\tau t - \theta) \quad (3.6)$$

where $\omega_{dop} = 2\pi f_{dop}$. Consider the following example to illustrate how a Doppler shift affects the range measurement of a return. For an RF carrier of 9.4 GHz and an LFM slope of 9e12

Hz/s, an object moving at 343 m/s (speed of sound) creates a range measurement error as follows

$$f_{dop} = \frac{2 \cdot 343 \cdot 9.4e9}{c} \quad (3.7)$$

$$f_{dop} = 21.5 \text{ kHz} \quad (3.8)$$

$$\text{range error} = 0.4m \quad (3.9)$$

The Doppler frequency range error for stretch processing can be derived as follows. Remember that the frequency from stretch processing corresponds to the delay of the target echo. Recall these Equations 1.14 and 1.17 from Chapter 1:

$$\Delta\tau = \frac{f_M}{\alpha}$$

$$\Delta R = \frac{c}{2} \Delta\tau$$

Substituting the Doppler frequency for f_M in the time delay equation and plugging the result into the range equation gives the range error from Doppler shifts.

$$\tau_{dop} = \frac{2v_r f_C}{c\alpha}$$

$$R_{dop} = \frac{c}{2} \frac{2v_r f_C}{c\alpha}$$

$$R_{dop} = \frac{v_r f_C}{\alpha} \quad (3.10)$$

where f_C is the RF carrier of the transmitted waveform. The error is inversely proportional to the slope of the LFM waveform. This is intuitive because as the slope increases, the each added hertz to the destretched frequency corresponds to a smaller increase in range.

The Doppler frequency can be extracted using digital processing and multiple pulses. This is possible because the ROC uses coherent processing. It is coherent because the same PLL is used to generate IF and RF for transmit and receive. If a target is stationary, the phase difference between the return and the LO should stay the same for each pulse. As the target moves, the phase difference will change also. There is ambiguity introduced if the target moves more than a wavelength between pulses.

3.3 Group Delay

Group delay, which is an indicator of phase distortion, is a possible source of error for stretch processing. It has the effect of smearing the target frequency in the frequency domain. Group delay (GD) is the rate of change of the phase with respect to the frequency. It has the unit of seconds. Wide bandwidth radar waveforms are especially susceptible to phase distortion. The equation for GD is

$$GD = -\frac{d\phi}{d\omega} \quad (3.11)$$

GD is used to measure the delay of a linear system block such as a filter. If the GD is constant, the system has a constant delay. This means that all frequencies of concern are delayed an equal amount. A constant GD implies a linear phase response. Do not mistake GD with transient analysis or the settling time of a system block. GD is a measure of how frequencies are delayed, not how long it takes a system block to settle after turning on.

Consider an ideal transmission line as an example to understand the concept of GD.

$$e^{-j\omega T} \quad (3.12)$$

The phase for the ideal transmission line is $-\omega T$, so the GD is a constant T. The GD is determined by the length of the transmission line. As the length of the transmission line

increases, the GD also increases. Non-constant GD means that some frequencies are delayed differently. This is common in filters for frequencies around the cutoff frequency.

With the preceding explanation of GD, there should be some intuition about how GD can affect stretch processing. Constant GD is not much of a concern. If the constant GD is large enough, it will add a range error. If the GD is not a constant, it will spread the power of the destretched frequency in the frequency domain, therefore, reducing the range resolution. Quantifying how GD affects stretch processing is possible because of the relationship between frequency and time. Consider a destretched frequency with frequency dependent phase distortion $\phi(\omega)$ destretched with it. How the phase distortion affected the destretching needs to be determined.

$$\cos(2\pi\alpha\Delta\tau t + \phi(\omega)) \quad (3.13)$$

The effect of the phase distortion is not immediately obvious since it is a function of frequency. The instantaneous frequency of this signal is

$$f_M = \alpha\Delta\tau + \frac{1}{2\pi} \frac{d\phi(\omega)}{dt} \quad (3.14)$$

The frequency dependent distortion has the following relationship to time during stretch processing

$$\omega = 2\pi\alpha t \quad (3.15)$$

To find the effect of the GD on the destretched frequency, time can be replaced with its relationship to frequency in Equation 3.14.

$$f_M = \alpha\Delta\tau + \alpha \frac{d\phi(\omega)}{d\omega} \quad (3.16)$$

$$f_M = \alpha(\Delta\tau + \text{GD}) \quad (3.17)$$

If GD is not constant, it will vary with time for the duration of the pulsewidth. If the frequency varies, the range resolution decreases. The range equation can be written with the GD effects included.

$$\begin{aligned}\Delta r &= \frac{f_M c}{\alpha} \frac{c}{2} \\ \Delta r &= (\Delta\tau + \text{GD}) \frac{c}{2}\end{aligned}\tag{3.18}$$

3.3.1 GD Variation

From the previous analysis, constant group delay is not detrimental to the radar system performance. If the ROC introduces a significant amount of constant group delay, it can be corrected through digital processing. Group delay variation is a much more detrimental issue. Group delay variation is non-linear phase distortion. Consider the following second order phase distortion ϕ :

$$\phi = a\omega^2 + b\omega\tag{3.19}$$

The constant coefficient a determines the effect of group delay variation on the signal. Translating the effect of a on the signal in the time domain is done as follows:

$$\begin{aligned}GD &= -\frac{d\phi}{d\omega} \\ &= -2a\omega - b\end{aligned}$$

Because of the relationship between time and frequency, ω can be rewritten in terms of time. After the substitution, the equation is in terms of instantaneous frequency.

$$\begin{aligned}GD &= -2a(2\pi\alpha t) - b \\ &= -4\pi a\alpha t - b\end{aligned}\tag{3.20}$$

The instantaneous frequency of the GD variation looks very similar to the instantaneous frequency of a LFM waveform. Indeed, it is a LFM waveform with a much smaller slope. The original α is scaled down by multiplication with a . This equation can be substituted for the GD in Equation 3.17 to get the result for GD variation.

$$f_M = \alpha(\Delta\tau - 4\pi a\alpha t - b) \quad (3.21)$$

The previous derivation of the effect phase distortion through GD may be a slightly convoluted route. The second order phase distortion can be directly included in the destretched waveform. The same result is obtained either way.

$$\begin{aligned} s(t) &= \cos(2\pi\alpha \Delta\tau t + \phi(\omega)) \\ &= \cos(2\pi\alpha \Delta\tau t + a\omega^2 + b\omega) \\ &= \cos(2\pi\alpha \Delta\tau t + a(2\pi\alpha t)^2 + b(2\pi\alpha t)) \\ &= \cos(2\pi\alpha (\Delta\tau + a2\pi\alpha t + b)t) \end{aligned} \quad (3.22)$$

3.3.2 GD Simulation

The previously derived effect of constant and linear GD can be verified through simulation. The phase distortion can be added to the stretch processing in both the time and frequency domain. The results should be the same and match the equations. Stretch processing can be achieved in the frequency domain by convolving the received and reference waveforms. Computationally, this is not nearly as efficient as multiplying in the time domain.

The phase distortion is added to the signal in the frequency domain multiplying the Fourier transform of the time signal with a complex exponential. The Fourier transform properties and pairs needed for simulating GD in the frequency domain are (\otimes is the symbol

for convolution)

$$\cos(\omega_0 t) \Leftrightarrow \pi [\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] \quad (3.23)$$

$$\cos(\omega_0 t + \phi) \Leftrightarrow \pi [e^{-j\phi}\delta(\omega + \omega_0) + e^{j\phi}\delta(\omega - \omega_0)] \quad (3.24)$$

$$x(t)v(t) \Leftrightarrow X(\omega) \otimes V(\omega) \quad (3.25)$$

Before simulation, knowing the bandwidth of the signal with GD will be helpful. Also note that the simulation assumes that the linear GD is constant across the whole waveform bandwidth. In reality, the linear GD will be worse around the edges of the signal BW. The bandwidth of the GD signal can be estimated by the non-constant part of Equation 3.21. Where T_{chirp} is the pulsewidth of the LFM, α is the slope of the LFM, and a is the constant coefficient from Equation 3.19, the bandwidth is

$$BW = 4\pi a \alpha^2 T_{chirp} \quad (3.26)$$

The parameters for the GD simulation are given in Table 3.2. Using the parameters in the table with Equation 3.26, the expected BW result of the simulation is 2.44 MHz. The undistorted result is a frequency at 9 MHz. The constant GD, b , causes an offset of 900 kHz.

Table 3.2: GD Simulation Parameters

Parameter	Value	Description
T_{chirp}	16.666 μ s	pulsewidth
α	9E12 $\frac{Hz}{s}$	slope of LFM
$\Delta\tau$	1E-6 s	time delay of received signal
a	1.44E-16	2 nd order phase distortion coefficient
b	0.1 E-6	1 st order phase distortion coefficient

The simulation results show that the GD added in the frequency domain and the GD added in the time domain are nearly the same. There is only a barely perceptible difference in the ripple of the destretched signals. Figure 3.2 shows a plot of the results. The plot on the left is the results found in the time domain, and the left plot is the results from the

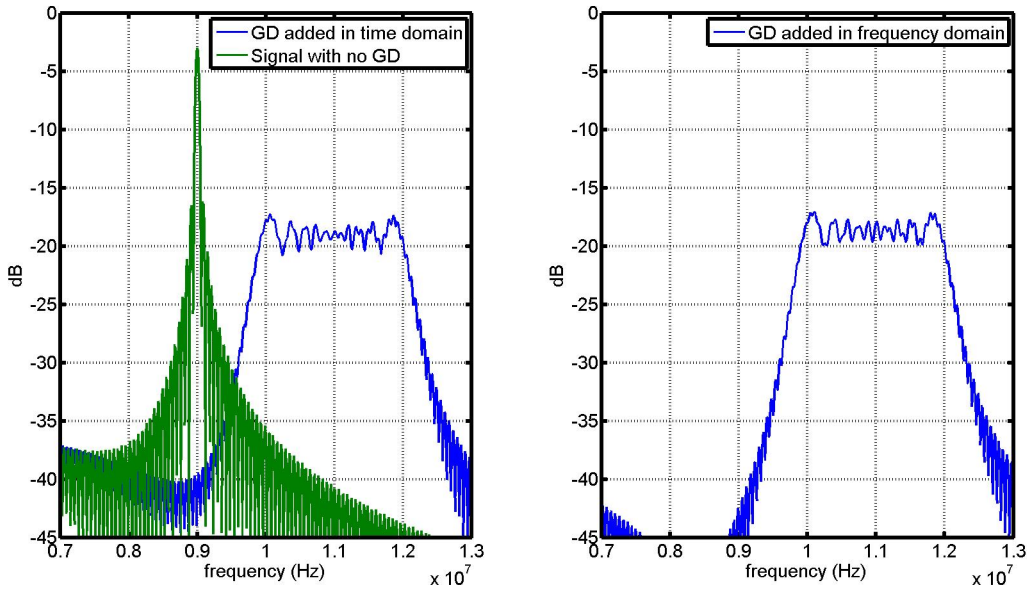


Figure 3.2: GD Verification

frequency domain simulation. The green plot on the right graph is the destretched signal with no GD. The 3-dB bandwidth of the GD signal is about 2.34 MHz. This is slightly less than predicted. There are a few reasons for this. Firstly, the actual bandwidth of an LFM waveform is about 0.98 of the frequency range covered by the LFM waveform. Secondly, the time bandwidth product of the LFM waveform produced by the linear GD is not $\gg 1$.

3.3.3 GD Power Loss

Figure 3.2 illustrates another problem that GD creates. The peak power in the waveform with GD is much lower than the signal with no GD. The difference in the figure is about 16 dB. The total energy of the signals is the same. The energy in the signal with no GD is concentrated at a single frequency; whereas, the energy in the waveform with linear GD is spread across a BW. This can make a significant difference when trying to detect low SNR target returns.

Sine wave pulses and LFM pulses have approximately the same average power. The equation to find the average power of a pulse of length T is

$$P_{avg} = \frac{1}{T} \int_0^T s(t)^2 dt \quad (3.27)$$

The average power of sine wave can be calculated with Equation 3.27 or found in any number of electrical engineering text books. The average power of a sine wave with amplitude A is $\frac{A^2}{2}$. The average power for an LFM waveform does not have a closed form solution, but it can be approximated

$$\begin{aligned} P_{avg} &= \frac{1}{T} \int_0^T (A \cos(\pi\alpha t^2))^2 dt \\ &= \frac{1}{T} \int_0^T A^2 \frac{(1 + \cos(2\pi\alpha t^2))}{2} dt \\ &= \frac{1}{T} \left[\int_0^T \frac{A^2}{2} dt + \int_0^T \frac{A^2 \cos(2\pi\alpha t^2)}{2} dt \right] \\ &= \frac{A^2}{2} + \frac{A^2 C(2\sqrt{\alpha}t)}{4\sqrt{\alpha}} \Big|_0^T \end{aligned}$$

$C(x)$ is a Fresnel integral. The integral converges to $\frac{1}{2}$ as x approaches infinity. For $x > 0$ All of the values of $C(x)$ are between 0 and 0.8, and for $x > 3$, the values stay between 0.4 and 0.6. For GD variation and LFM waveforms, $\alpha \gg 1$. This means that the Fresnel integral can be approximated as $\frac{1}{2}$ and dividing it by the $\sqrt{\alpha}$ makes it a very small number. This means that the average power in a sine wave and an LFM wave of equal amplitude is equal.

$$P_{avg}^{LFM} = P_{avg}^{sine} = \frac{A^2}{2} \quad (3.28)$$

Since the total energy is the same for a waveform with and without GD variation, the gain-bandwidth product should be the same for each waveform as long as the amplitude is equal. For an ideal sine wave the bandwidth is infinitely small. In, practice this is not the case. The sine wave pulse is a sine wave multiplied by a rectangle function. In the frequency

domain, the sine wave pulse, is a delta function convolved with a sinc function. The sinc function determines the bandwidth of the pulse. If the main lobe of the sinc is approximated as a rectangle, and the pulse affected by group delay is approximated as a rectangle, the following relationship is true:

$$\text{BW}_{sine} P_{peak}^{sine} = \text{BW}_{GD} P_{peak}^{GD} \quad (3.29)$$

Refer to Figure 3.2 as an example. The 3-dB bandwidth of the sinc function is 55 kHz and the peak power is -3 dB. The bandwidth of the GD pulse is about 2.34 MHz. Using Equation 3.29, the peak power of the GD pulse should be about -19 dB—this is confirmed by examining the plot.

3.3.4 GD Summary

The effect that GD has on both power level and signal bandwidth has been established. The best way to compile the analysis into useful design equations is to look at how much it varies from the start to the finish of an LFM bandwidth. If the variation is not linear, use the difference between the highest and lowest GD values. This difference is referred to as ΔGD . This strategy is useful because design software such as Cadence has the ability to calculate and plot GD. If the GD is not directly available, export the phase of the system and perform a numerical derivative using software such as MATLAB. ΔGD can be found using Equation 3.20. Substitute T_{chirp} for t . Once ΔGD is found, the bandwidth and peak power loss can be calculated for a given LFM slope, α :

$$\text{BW}_{GD} = \alpha \Delta\text{GD} \quad (3.30)$$

$$P_{loss} = 10 \log \left(\frac{1}{T_{chirp} \text{BW}_{GD}} \right) \quad (3.31)$$

3.3.5 GD Range Resolution Effect

The range resolution is reduced by the Δ GD. As a target return is spread out in the frequency domain, it may overlap with a nearby return, making the targets indistinguishable. Recall the original range resolution equation, which is inversely proportional to the bandwidth of the pulse:

$$\Delta R = \frac{c}{2BW}$$

The bandwidth of an LFM waveform is the slope α multiplied with the pulsewidth of the waveform. Substitute the bandwidth of the LFM waveform with the bandwidth in the range resolution equation to get the following:

$$\Delta R = \frac{c}{2\alpha T_{chirp}} \quad (3.32)$$

The relative range equation for stretch processing can be written in terms of the destretched frequency as

$$\text{range} = \frac{c f_M}{2 \alpha}$$

where f_M is the destretched frequency. If the equation is divided by the received frequency on each side of the equal sign, it represents meters per Hz. In other words, it represents how many meters correspond to a 1 Hz change in the destretched frequency. It is the relationship or mapping between range and frequency in stretch processing.

$$\frac{\text{range}}{f_M} = \frac{c}{2\alpha} \quad (3.33)$$

Notice that the right hand side of Equation 3.33 is part of Equation 3.32, the range resolution equation for stretch processing. Equation 3.32 is comprised of two parts: the meters per Hz multiplied by $\frac{1}{T_{chirp}}$. $\frac{1}{T_{chirp}}$ is the bandwidth of the destretched frequency. Think of the pulsewidth of the chirp as a rectangle function multiplying the destretched frequency. This rectangle function transforms to a sinc in the frequency domain with a bandwidth of $\frac{1}{T_{chirp}}$.

The range resolution for stretch processing is just the bandwidth dependent on the pulsewidth multiplied with a constant that maps the bandwidth to meters. Group delay variation spreads the bandwidth of the destretched frequency. Once the bandwidth of the linear GD variation is larger than $\frac{1}{T_{chirp}}$, it can be used in the range resolution equation. Equation 3.26 or Equation 3.31 can be used to calculate the bandwidth of the GD affected signal. The modified range resolution equation is

$$\Delta R = \begin{cases} \frac{c}{2\alpha} \frac{1}{T_{chirp}} & \text{if } \frac{1}{T_{chirp}} > BW_{GD} \\ \frac{c}{2\alpha} BW_{GD} & \text{if } \frac{1}{T_{chirp}} < BW_{GD} \end{cases} \quad (3.34)$$

Equation 3.34 gives some idea of what may be an acceptable amount of GD variation for a system. For BW_{GD} less than $\frac{1}{T_{chirp}}$, the effects may be tolerable. Small amounts of GD variation can still cause a frequency shift. For example, $\frac{1}{T_{chirp}} = BW_{GD} = 60\text{kHz}$. The bandwidth of the signal affected with GD is not much larger than an undistorted signal. The center of the signal is now shifted from the ideal received frequency. f_M to $f_M + \frac{BW_{GD}}{2}$.

Figure 3.3 is an example of GD variation causing two targets to appear as one. The plot on the left shows the two returns with no GD variation. The frequencies are separated by 540 kHz. The plot on the right shows the same two returns with $BW_{GD} = 600$ kHz. The two returns are indistinguishable.

3.4 IQ Imbalance

The stretch processing for the ROC is implemented with two channels. The received baseband signal is split into two channels before stretch processing. Each channel destretches the baseband signal. The destretch signal for one channel is $\cos(\alpha\pi t^2)$ and is labeled the in-phase or **I** channel. The other destretch signal is offset by 90° and is labeled the quadrature-phase or **Q** channel. The quadrature signal can be written as $\sin(\pi\alpha t^2)$.

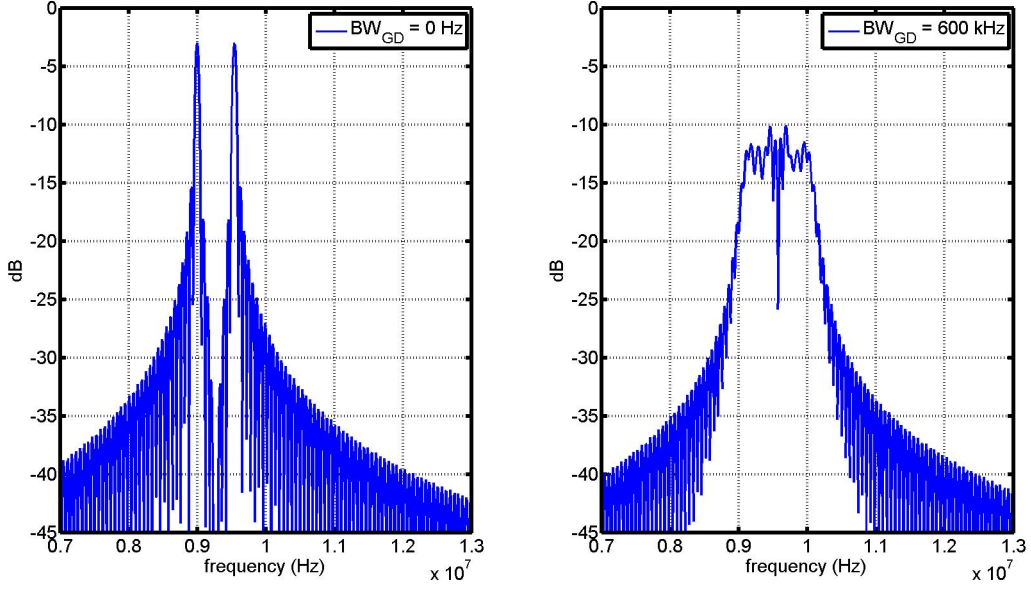


Figure 3.3: The effect of ΔGD on range resolution

After the stretch processing, each channel has a baseband variable gain amplifier, a lowpass filter, and a mixer before being combined. Ideally, each channel is identical with the exception of the phase offset in the initial stretch processing. Unfortunately, due to process variations and circuit layout inequalities, the channels are not equal. The channel inequalities are modeled as an amplitude and phase difference. Channel inequalities have two effects on the processed signal: The expected frequency loses some power, and most importantly, an image frequency appears in the output spectrum of the processed signal.

To begin to analyze IQ channel imbalances, start with the ideal channel outputs for a single target. **I Mixer:**

$$s_I = \cos(\omega_{offset}t + 2\pi\alpha(\Delta\tau)t) + \cos(\omega_{offset}t - 2\pi\alpha(\Delta\tau)t) \quad (3.35)$$

Q Mixer:

$$s_Q = -\cos(\omega_{offset}t + 2\pi\alpha(\Delta\tau)t) + \cos(\omega_{offset}t - 2\pi\alpha(\Delta\tau)t) \quad (3.36)$$

Ideally, the two channels are subtracted to entirely cancel the $f_{offset} - \alpha\Delta\tau$ frequency term. Because of IQ imbalances, this frequency is not entirely canceled. If the image frequency is strong enough, it will appear as a false return in the digital processing. As Equations 3.36 and 3.35 show, the outputs of the two channels are not actually in quadrature phase when they are combined. The channels are more accurately described as quadrature stretch processing used to cancel an image frequency. The term “IQ imbalance” is used to refer to the phase and amplitude inequalities that occur because of the two signal paths.

3.4.1 Phasor Representation

The easiest way to derive the effects of IQ imbalances on the final frequency spectrum is to treat the frequency terms as phasors. Using Euler’s formula, a cosine wave can be represented with complex exponentials:

$$A \cos(\omega t + \theta) = A \frac{e^{j(\omega t + \theta)} + e^{-j(\omega t + \theta)}}{2} \quad (3.37)$$

$$= \text{Re}\{Ae^{j(\omega t + \theta)}\} \quad (3.38)$$

When dealing with linear combinations of sinusoids of the same frequency, all of the terms share $e^{j\omega t}$. This term can be dropped during analysis, and each sinusoid can be represented with a magnitude and phase term. This is called phasor representation. The magnitude is always positive. A negative number is represented by changing the phase of the phasor.

3.4.2 Linear Combination

Consider the summation of two sinusoids with the same frequency but different phases and amplitudes.

$$A \cos(\omega t + \theta) + B \cos(\omega t + \phi) = \text{Re}\{Ae^{j(\omega t + \theta)}\} + \text{Re}\{Be^{j(\omega t + \phi)}\} \quad (3.39)$$

$$= C \cos(\omega t + \beta) \quad (3.40)$$

Equation 3.40 reduces to a cosine wave of the same frequency but different amplitude and phase. The magnitude of the wave can be found using phasor representation and the following property: The square root of a complex number multiplied by its complex conjugate is the magnitude.

$$\begin{aligned}
C &= \sqrt{(Ae^{j\theta} + Be^{j\phi}) \cdot (Ae^{-j\theta} + Be^{-j\phi})} \\
&= \sqrt{A^2 + AB e^{j(\theta-\phi)} + AB e^{-j(\theta-\phi)} + B^2} \\
&= \sqrt{(A^2 + B^2 + 2AB \cos(\theta - \phi))} \tag{3.41}
\end{aligned}$$

If the phases are equal, the magnitude C is simply $A + B$. If the phases are not equal, the magnitude will always be less than $A + B$. If the phases are 180° out of phase, the magnitude reduces to $C = A - B$; the cosines are subtracted.

The phase β of Equation 3.40 is a function of both the magnitudes and phases of the cosines being combined. For any complex number $x + jy$, the phase is $\arctan(\frac{y}{x})$. The sum of the cosines can be separated into the real and imaginary parts to find the phase:

$$\begin{aligned}
Ae^{j\theta} + Be^{j\phi} &= A(\cos(\theta) + j \sin(\theta)) + B(\cos(\phi) + j \sin(\phi)) \\
&= A \cos(\theta) + B \cos(\phi) + j(A \sin(\theta) + B \sin(\phi))
\end{aligned}$$

The phase can now be found as

$$\beta = \arctan \left(\frac{A \sin(\theta) + B \sin(\phi)}{A \cos(\theta) + B \cos(\phi)} \right) \tag{3.42}$$

The cosine terms that are supposed to constructively add when combining the channels from stretch processing are not as susceptible to IQ imbalances as the terms that are supposed to cancel. The $\omega_{offset}t + 2\pi\alpha(\Delta\tau)t$ may lose some power or gain power because of amplitude variations. Phase variations will always cause a reduction in power. If the amplitudes are

the same, the effect of a phase difference on the amplitude can be calculated as follows:

$$C = 2A \cos\left(\frac{\theta}{2}\right) \quad (3.43)$$

The terms that are supposed to cancel at the combination of the two channels are the ones that need to be balanced. Discrepancies between the channels will cause an unwanted image frequency at $f_{offset} - \alpha\Delta\tau$. The location of the image frequency is always the correct frequency mirrored around the offset frequency. For example, an offset frequency of 10 MHz and a destretched frequency of 1 MHz creates an image frequency at 19 MHz. To simplify the analysis, let one of the channels be normal or ideal. The other channel has the phase and amplitude variation. To accomplish this, set $B = 1$ and $\phi = 0$ in Equation 3.40. The equation for the linear combination of two cosines can now be written as

$$-A \cos(\omega t + \theta) + \cos(\omega t) = \sqrt{A^2 + 2A \cos(\theta + \pi) + 1} \cos\left(\omega t + \arctan\left(\frac{A \sin(\theta + \pi)}{A \cos(\theta + \pi) + 1}\right)\right) \quad (3.44)$$

Since the two cosines are going to be subtracted, π needs to be added to θ . This accounts for the negative amplitude of A . For determining how much power the image frequency has, the phase of the cosine on the right hand side of Equation 3.44 is not important. The power of the image frequency is

$$P_{image} = \frac{A^2 - 2A \cos(\theta) + 1}{2} \quad (3.45)$$

Figure 3.4 is a contour plot showing the power of the image frequency for phase and amplitude variation. The y-axis is the amplitude variation in dB. The x-axis is the phase variation in degrees. The contours are also in dB. The power of the image frequency is approximately equal to the amplitude variation for small values of phase variation.

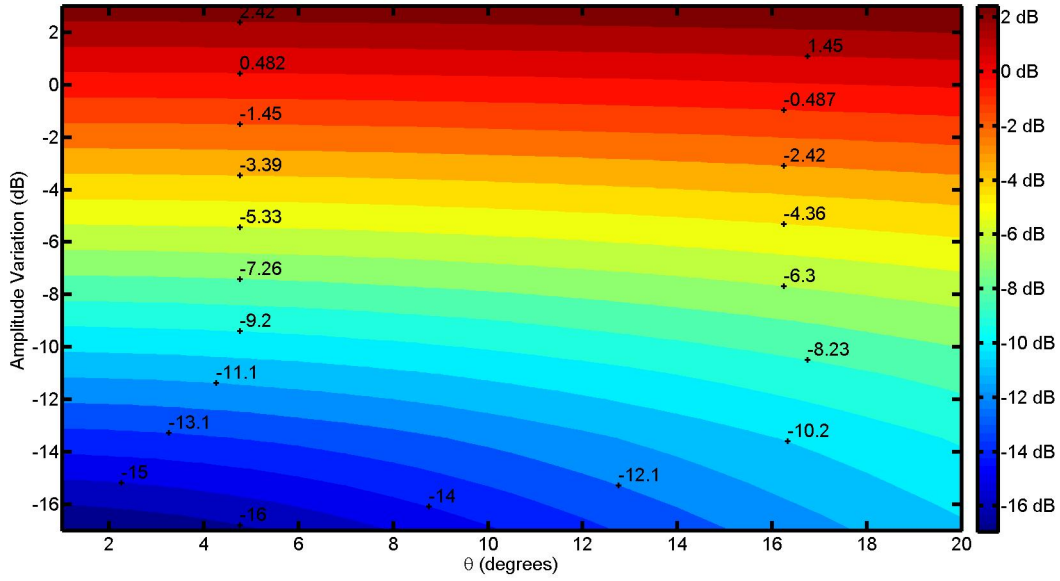


Figure 3.4: Contour plot for the power of the image frequency

3.5 Processing Gain

Processing gain is a term that describes a system's ability to reject interference. In spread spectrum systems, processing gain is the ratio of the spread bandwidth to the baseband bandwidth. This immediately gives an idea of what processing gain means for stretch processing. To make it analogous to spread spectrum, the LFM waveform is the spread signal. The energy of the signal is spread across a wide bandwidth. After stretch processing, this energy is compressed into a single frequency. Intuitively, the processing gain for stretch processing is going to be related to the ratio of signal bandwidth before stretch processing to after stretch processing. Assuming a rectangular window, the bandwidth after stretch processing is $\frac{1}{T_{chirp}}$. From this reasoning, processing gain should be proportional or equal to BWT_{chirp} . This number has already been seen in Chapter 1 as the time-bandwidth product. More generally, the processing gain G_p can be defined as the ratio of the output SNR to the input SNR:

$$G_p = \frac{SNR_o}{SNR_i} \quad (3.46)$$

3.5.1 Matched Filter

Matched filters are relevant to the discussion of processing gain. A matched filter is designed to have a maximum output SNR. A matched filter convolves a noisy received signal with the conjugated, time-reversed version of the expected signal. The SNR at the output of a matched filter is

$$\text{SNR}_{mf} = \frac{E}{N_0} \quad (3.47)$$

where E is the energy of the signal and N_0 is the one-sided noise power spectrum. Notice that the signal power is only dependent on the energy in the signal. Since a matched filter is ideal, its output SNR can serve as a benchmark to other filters and processing techniques. *The Radar Handbook* [1] defines a filter matching loss as

$$L_m = \frac{\text{SNR}_{mf}}{\text{SNR}_0} \quad (3.48)$$

This equation will be useful for creating a figure of merit for the stretch processing implementation of the ROC. According to [3], the processing gain for an LFM waveform with a matched filter is the time-bandwidth product. This matches up with the previous speculation that the processing gain of stretch processing is equal or proportional to the time-bandwidth product. The authors of [2] claim that the filter loss from stretch processing is equal to or less than matched filtering depending on the noise bandwidth.

3.5.2 Processing Gain Derivation

Although the authors of [2] claim to have derived the filter loss for stretch processing, a thorough explanation and derivation of the processing gain for stretch processing seems to be lacking in available literature. The goal of this section is to provide an intuitive, easy-to-follow explanation as well as the derivation.

Consider a received signal in the form of

$$s(t) = \text{rect}\left(\frac{t - \tau}{T_{chirp}}\right) e^{(j\pi\alpha(t-\tau)^2)} \quad (3.49)$$

The complex exponential is used instead of a cosine merely out of convenience. The math is much easier to follow. Input noise $n(t)$ is added to the signal. Assume that the noise is band-limited white Gaussian noise. The noise is described by its power spectrum density N_0 and the system noise bandwidth B_n . The system noise bandwidth should be $B_n \geq B_s$ where B_s is the bandwidth of the received LFM waveform. Most systems should have a noise bandwidth very close to the bandwidth of the received signal. For the sake of completeness, they are assumed to be different. The noise power $P_n = N_0 B_n$. The signal plus the noise is the input into the correlation mixer.

The average power of an LFM waveform is derived in Section 3.3.3 and Equation 3.28 as $\frac{A^2}{2}$. Since a complex exponential is used here instead of a cosine, the average power is simply A^2 . The SNR of the input waveform can now be calculated as

$$SNR_i = \frac{A^2}{B_s N_0} \quad (3.50)$$

The reference signal is also in the form of a complex exponential:

$$r(t) = \text{rect}\left(\frac{t}{T_r}\right) e^{(-j\alpha\pi t^2)} \quad (3.51)$$

The reference signal has a time delay of 0, and a bandwidth $B_r \geq B_s + 2\alpha\tau_{max}$. The reference signal and the received signal are multiplied together to implement stretch processing.

$$\begin{aligned} y(t) &= (s(t) + n(t))r(t) \\ &= \text{rect}\left(\frac{t - \tau}{T_{chirp}}\right) e^{j(-2\pi\alpha\tau(t-\frac{\tau}{2}))} + n(t)r(t) \end{aligned} \quad (3.52)$$

The first part of Equation 3.52 is simply the result of stretch processing. The noise multiplied by the reference is of interest though. First the Fourier transform of $y(t)$ can be found.

$$Y(f) = AT_{chirp}e^{j\pi\alpha\tau^2}\text{sinc}((f + \alpha\tau)T_{chirp}) + n_{out}(f) \quad (3.53)$$

The Fourier transform is a sinc function centered around the destretched frequency. This is exactly what is expected from earlier discussions of stretch processing. The important thing to realize here is the power at the frequency $-\alpha\tau$:

$$P_{out} = (AT_{chirp})^2 \quad (3.54)$$

The last piece missing from the processing gain equation is the output noise power. Since $n_{out}(f)$ is the Fourier transform of $n(t)r(t)$, the frequency domain solution is the convolution of the reference signal with the noise. The expected value of the noise power at a given frequency is the convolution of the noise power spectrum with the power spectrum of the reference signal. The noise power spectrum is given simply by

$$P_n(f) = N_0\text{rect}\left(\frac{f}{B_n}\right) \quad (3.55)$$

The magnitude squared or power of the reference signal spectrum is approximated with a scaled rectangle function.

$$|R(f)|^2 = \frac{1}{\alpha}\text{rect}\left(\frac{f}{B_r}\right) \quad (3.56)$$

Notice that the convolution is going to be of two scaled rectangle functions. This is an interesting result. Recall that the convolution of two equal rectangle functions is a triangle. If the rectangle are of different sizes, the triangle becomes an isosceles trapezoid. In other words, the triangle gets a flat top with a size dependent on how big the difference between the rectangles is. This means that in some situations the output SNR can be dependent on the destretched frequency. The convolution needs to be solved before the ramifications are

discussed any further.

$$\begin{aligned}
E[n_{out}(f)] &= |R(f)|^2 \otimes P_n(f) \\
&= \int_{-\infty}^{\infty} |R(f' - f)|^2 P_n(f') df' \\
&= \frac{N_0}{\alpha} \int_{-\infty}^{\infty} \text{rect}\left(\frac{f' - f}{B_r}\right) \text{rect}\left(\frac{f'}{B_n}\right) df'
\end{aligned}$$

Before the solution to the convolution is presented, some limits on the problem can simplify it. The only frequencies of interest are the ones that can correspond to a received time delay. Equation 3.1 can be multiplied by α and solved for f_{cutoff} to set the upper limit of the convolution solution as $\frac{B_r - B_s}{2}$.

$$E[n_{out}(f)] = \begin{cases} \frac{N_0}{\alpha} B_r & \text{if } |f| < \frac{B_n - B_r}{2} \\ \frac{N_0}{\alpha} B_n & \text{if } |f| < \frac{B_r - B_n}{2} \\ \frac{N_0}{\alpha} \frac{B_r + B_n}{2} - f & \frac{|B_n - B_r|}{2} \leq |f| \leq \frac{B_r - B_s}{2} \end{cases} \quad (3.57)$$

Now that the noise power is calculated, the output SNR can be calculated. The output SNR is dependent on the noise bandwidth. Using Equations 3.56 and 3.57, the output SNR is

$$SNR_{out} = \begin{cases} \frac{A^2 T_{chirp} B_s}{N_0 B_r} & \text{if } |f| < \frac{B_n - B_r}{2} \\ \frac{A^2 T_{chirp} B_s}{N_0 B_n} & \text{if } |f| < \frac{B_r - B_n}{2} \\ \frac{2A^2 T_{chirp} B_s}{N_0 (B_n + B_r - |2f|)} & \frac{|B_n - B_r|}{2} \leq |f| \leq \frac{B_r - B_s}{2} \end{cases} \quad (3.58)$$

This is an interesting result. In some instances, the output SNR is dependent on the de-stretched frequency. In other words, the output SNR is dependent on the time delay of the received signal. The important observation, which was clear from the beginning, is that the noise bandwidth of the system needs to be as close to the bandwidth of the received LFM waveform as possible.

Calculating the processing gain is now a straightforward step. The output SNR divided by the input SNR is the processing gain.

$$G_p = \begin{cases} \frac{T_{chirp} B_s B_s}{B_r} & \text{if } |f| < \frac{B_n - B_r}{2} \\ \frac{T_{chirp} B_s B_s}{B_n} & \text{if } |f| < \frac{B_r - B_n}{2} \\ \frac{2T_{chirp} B_s B_s}{(B_n + B_r - |2f|)} & \frac{|B_n - B_r|}{2} \leq |f| \leq \frac{B_r - B_s}{2} \end{cases} \quad (3.59)$$

The processing gain is the same as a matched filter for $B_s = B_n$. If the noise bandwidth is larger than the reference signal bandwidth, the degradation in the processing gain is proportional to B_r/B_s . If the noise bandwidth is smaller than the reference signal bandwidth, the degradation in the processing gain is proportional to B_n/B_s .

3.5.3 ROC Processing Gain

The processing gain of the ROC is not ideal. The system is sub-par not because of hardware and process variations. It is not ideal because of the design itself. The problem is rooted in the image frequency that is the result of stretch processing. Recall that stretch processing produces the single-tone difference frequency and the sum frequency, which will be termed the chirp-squared image. The chirp-squared image is another LFM waveform with a bandwidth twice as large as the original. If the reference signal delay is set to zero, the chirp squared term from the ROC can be written as

$$chirp^2 = \cos(2\pi\alpha(t^2 - \tau t + \frac{\tau^2}{2})) \quad (3.60)$$

Its presence in the spectrum limits the output SNR of any received signal to the processing gain. For example, consider an input signal with an SNR of 5 dB and a processing gain of 34 dB. The output SNR should be 39 dB, but if the chirp-squared term is present in the spectrum, the output frequency is only 34 dB (the processing gain) above the chirp-squared term. This may not seem like a problem at first. All it does to an input signal is limit the

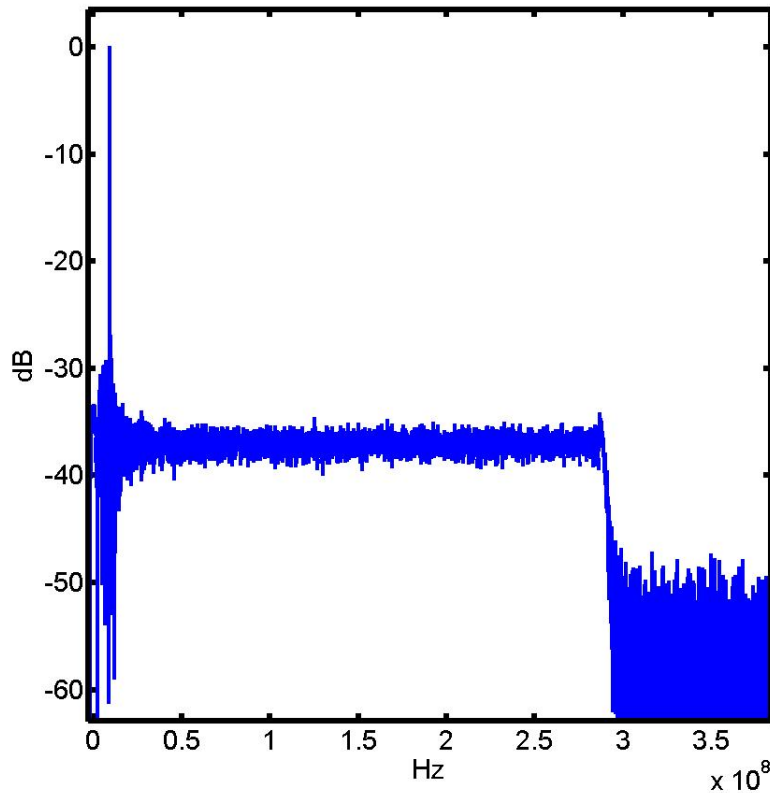


Figure 3.5: Plot showing chirp-squared term

SNR to the processing gain. Input signals that are below the noise floor are not affected at all. Unfortunately, the chirp-squared term causes another major problem. It shields other signal returns with less power. With the previous example, any signal with an SNR below 5 dB after stretch processing will be hidden by the chirp-squared term. As an example, Figure 3.5 shows the spectrum of a destretched return with the single tone and the chirp-squared term. The received signal has an SNR of 10 dB. The processing gain is ideally around 34 dB. The output SNR should be 44 dB, but it is 34 dB because of the chirp-squared term. Notice that at 300 MHz the chirp-squared term ends. This is because the chirp-squared bandwidth is twice the original bandwidth of 150 MHz.

As an example of the chirp squared term hiding a smaller signal, a signal with -40 dB of power is added to the signal in Figure 3.5. The stretch processing is calculated for both

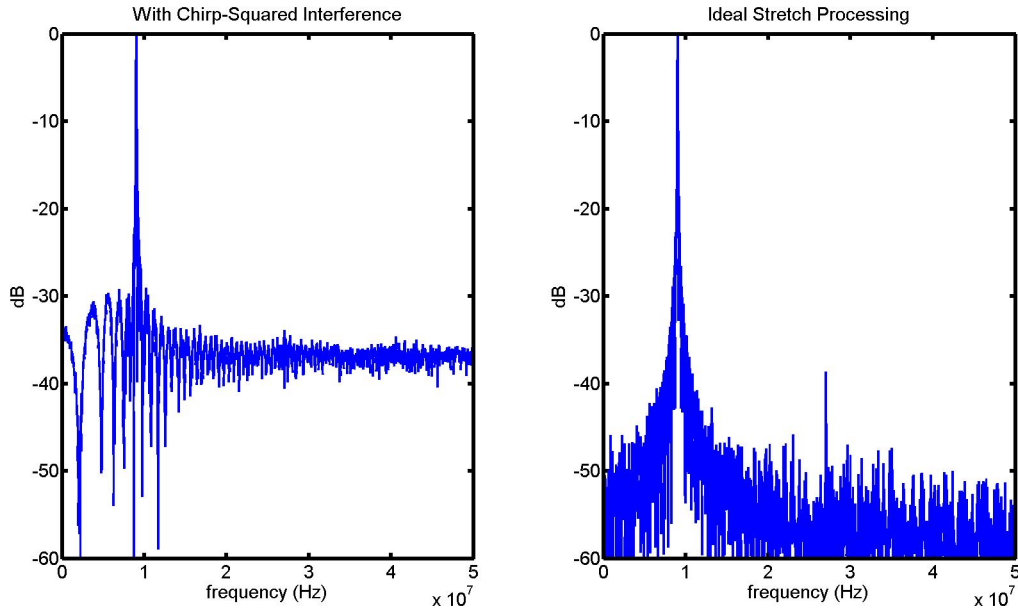


Figure 3.6: a)Chirp-squared term hiding a smaller tone b)Ideal stretch Processing

chirp-squared interference and ideal stretch processing. The left plot in Figure 3.6 is the chirp-squared term. There is no second tone visible. The plot on the right is ideal stretch processing. The low power frequency is visible, and the stronger frequency has a higher SNR than the plot on the left.

3.6 Correct Stretch Processing on ROC

The interference from the chirp-squared term is a serious flaw. In the initial design of the ROC, focus was centered on differentiating positive and negative ranges and using one ADC. The chirp-squared term was not forgotten about, but it was given secondary importance. After the first fabrication of the ROC, the system design was improved to remove the chirp-squared term.

Recall the building blocks that are the most costly for RFIC design. In any changes made to the system, extra filters, frequencies not easily available, and ADCs are to be avoided if possible. Adding mixers and summers are much more desirable.

There are two possibilities to get rid of the chirp-squared term. First, quadrature mixing can be used to cancel the image. Second, the stretch processing can be done at an intermediate frequency. This moves the chirp squared term to $f_{IF} + chirp^2$. As long as the IF is high enough to move the chirp-squared term completely out of the possible range of received frequencies, it can be filtered out. Stretch processing at IF is the best option for the ROC. Only one extra mixer and summer is added to the system. No new frequencies are needed.

In the original implementation of the ROC, the quadrature stretch processing is performed in the baseband. The quadrature stretch processing could not be performed at IF because no quadrature IF+LFM was available. The quadrature phase of the IF signal can be obtained from the available signals. Figure 3.7 is the updated stretch processing diagram. The diagram is not the whole transceiver. It only shows the blocks necessary for the stretch processing. The input signal that has already been down-converted to IF is injected at mixers M5 and M6.

3.6.1 Analysis

Using Figure 3.7, the system analysis for the stretch processing that avoids the chirp-squared interference can easily shown. For the analysis, the input and output of each relevant system block will be shown. The blocks will have an equation with the block input on the left side and the block output on the right side. The amplifiers and buffer is not shown because the signal frequency does not change. Following the analysis of stretch processing in Chapter 2, the rectangle functions are ignored throughout the analysis. They can be added to the final result without loss of accuracy as long as the received signal is assumed to be within the range-window. Mixers M1-M4 and summers S1 and S2 are used to generate the quadrature LFM waveforms at the IF. These waveforms are used for the stretch processing.

M1

$$\cos(\omega_{IF}t) \cdot \cos(\alpha\pi t^2) = \frac{1}{2} \cos(\omega_{IF}t + \alpha\pi t^2) + \frac{1}{2} \cos(\omega_{IF}t - \alpha\pi t^2) \quad (3.61)$$

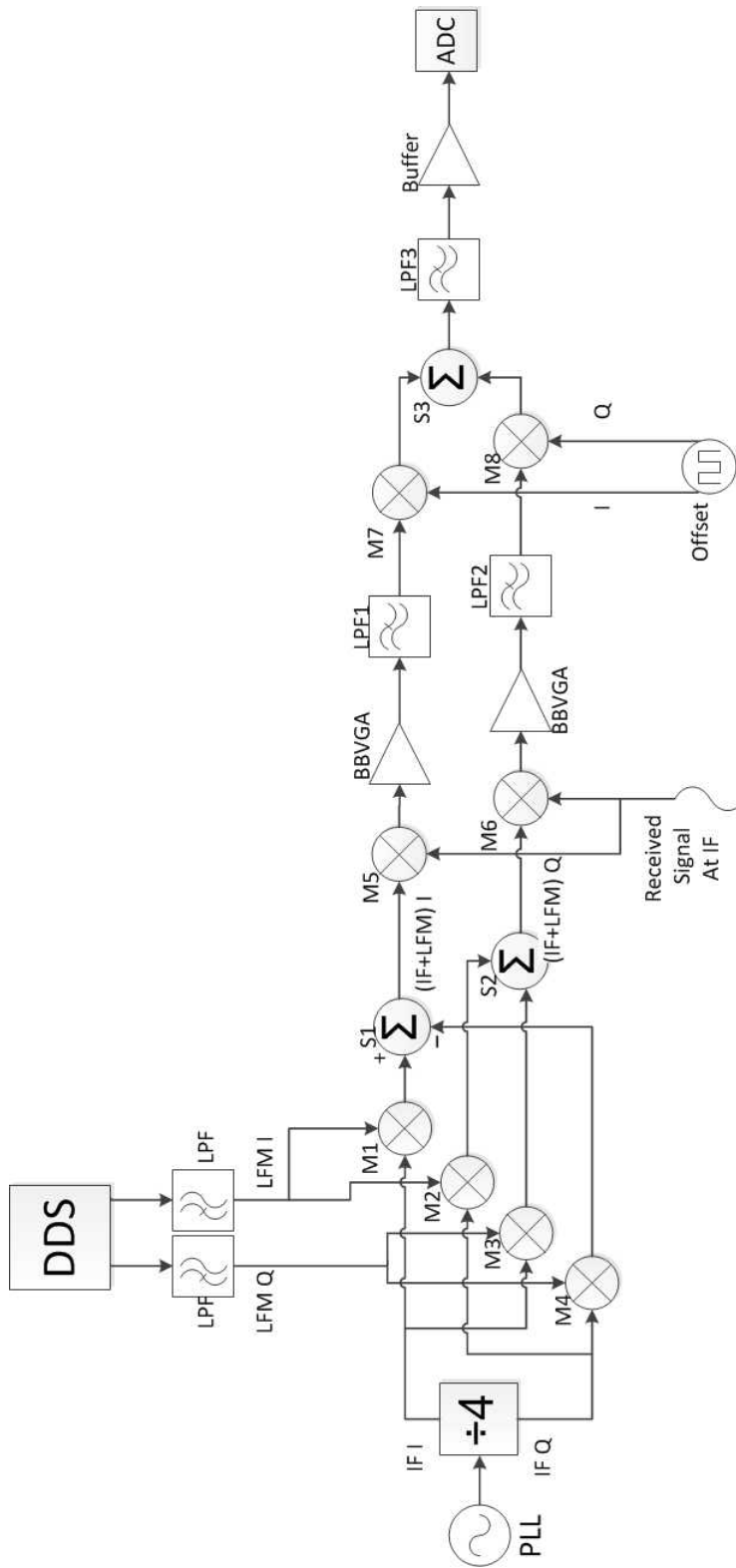


Figure 3.7: Updated Stretch Processing

M2

$$\sin(\omega_{IF}t) \cdot \cos(\alpha\pi t^2) = \frac{1}{2} \sin(\omega_{IF}t + \alpha\pi t^2) + \frac{1}{2} \sin(\omega_{IF}t - \alpha\pi t^2) \quad (3.62)$$

M3

$$\cos(\omega_{IF}t) \cdot \sin(\alpha\pi t^2) = \frac{1}{2} \sin(\omega_{IF}t + \alpha\pi t^2) - \frac{1}{2} \sin(\omega_{IF}t - \alpha\pi t^2) \quad (3.63)$$

M4

$$\sin(\omega_{IF}t) \cdot \sin(\alpha\pi t^2) = \frac{1}{2} \cos(\omega_{IF}t - \alpha\pi t^2) - \frac{1}{2} \cos(\omega_{IF}t + \alpha\pi t^2) \quad (3.64)$$

S1 is the difference between M1 and M4. The in-phase IF+LFM is generated at its output.

S1

$$\begin{aligned} & \frac{1}{2}(\cos(\omega_{IF}t + \alpha\pi t^2) + \cos(\omega_{IF}t - \alpha\pi t^2)) \\ & + \frac{1}{2}(\cos(\omega_{IF}t - \alpha\pi t^2) - \cos(\omega_{IF}t + \alpha\pi t^2)) \\ & = \cos(\omega_{IF}t\alpha\pi t^2) \end{aligned} \quad (3.65)$$

S2 is the sum of M2 and M3. The quadrature-phase IF+LFM is generated at its output.

S2

$$\begin{aligned} & \frac{1}{2}(\sin(\omega_{IF}t + \alpha\pi t^2) + \sin(\omega_{IF}t - \alpha\pi t^2)) \\ & + \frac{1}{2}(\sin(\omega_{IF}t + \alpha\pi t^2) - \sin(\omega_{IF}t - \alpha\pi t^2)) \\ & = \sin(\omega_{IF}t\alpha\pi t^2) \end{aligned} \quad (3.66)$$

Next the quadrature stretch processing is performed at M5 and M6. Remember that the received signal has already been down-converted to the IF. This is the step that offers improvement over the original stretch processing implementation in the ROC. The chirp-squared term is mixed up to $2\omega_{IF}$.

M5

$$\begin{aligned}
& \cos(\omega_{IF}t + \alpha\pi t^2) \cdot A \cos(\omega_{IF}t + \pi\alpha(t - \tau)^2) \\
&= \frac{A}{2} (\cos(2\omega_{IF}t + 2\pi\alpha(t^2 - \tau t + \frac{\tau^2}{2})) + \cos(\pi\alpha(2\tau t - \tau^2))) \tag{3.67}
\end{aligned}$$

M6

$$\begin{aligned}
& \sin(\omega_{IF}t + \alpha\pi t^2) \cdot A \cos(\omega_{IF}t + \pi\alpha(t - \tau)^2) \\
&= \frac{A}{2} (\sin(2\omega_{IF}t + 2\pi\alpha(t^2 - \tau t + \frac{\tau^2}{2})) + \sin(\pi\alpha(2\tau t - \tau^2))) \tag{3.68}
\end{aligned}$$

After the stretch processing the stretch processing, each channel is amplified and filtered. The filter cut-off frequency only needs to be as large enough to pass the frequency corresponding to the maximum time delay. The LPFs also remove the $2\omega_{IF}$ terms. After filtering each channel is mixed with an offset frequency. The offset frequency has to be at least the same as the filter cutoff frequency to move any negative frequencies into the positive range. The offset frequency has an I and Q output. This allows for the image to be canceled when the two channels are summed.

M7

$$\begin{aligned}
& \frac{A}{2} \cos(\pi\alpha(2\tau t - \tau^2)) 2 \cos(\omega_{off}t) \\
&= \frac{A}{2} (\cos(\pi\alpha(2\tau t - \tau^2) + \omega_{off}t) + \cos(\pi\alpha(2\tau t - \tau^2) - \omega_{off}t)) \tag{3.69}
\end{aligned}$$

M8

$$\begin{aligned}
& \frac{A}{2} \sin(\pi\alpha(2\tau t - \tau^2)) \cdot 2 \sin(\omega_{off}t) \\
&= \frac{A}{2} (\cos(\pi\alpha(2\tau t - \tau^2) - \omega_{off}t) - \cos(\pi\alpha(2\tau t - \tau^2) + \omega_{off}t)) \tag{3.70}
\end{aligned}$$

After the offset mixing, the two channels are subtracted to cancel the image frequency. Canceling this term through quadrature mixing is very important. It allows for the distinction between positive and negative relative ranges.

S3

$$\begin{aligned}
& \frac{A}{2}(\cos(\pi\alpha(2\tau t - \tau^2) + \omega_{offset}) + \cos(\pi\alpha(2\tau t - \tau^2) - \omega_{offset})) \\
& - \frac{A}{2}(\cos(\pi\alpha(2\tau t - \tau^2) - \omega_{offset}) - \cos(\pi\alpha(2\tau t - \tau^2) + \omega_{offset})) \\
& = A \cos(\pi\alpha(2\tau t - \tau^2) + \omega_{offset})
\end{aligned} \tag{3.71}$$

After the two channels are combined, the result is passed through an anti-aliasing filter before being buffered and sampled by the ADC. The anti-aliasing filter is necessary to remove any harmonics introduced during the mixing.

3.7 Interference

Much like matched filtering, stretch processing is tolerant to some interference. Whereas as the received LFM waveform is compressed in frequency, uncorrelated waveforms are stretched. Interference of a waveform with a similar slope will cause a problem though. The ideal processing gain for stretch processing is the time-bandwidth product. A received pulse has energy evenly spread across the bandwidth of the waveform. When the pulse is destretched the energy is compressed into a bandwidth equal to the inverse of the pulsewidth, $\frac{1}{T_{chirp}}$. Conversely, when a single-tone pulse is multiplied with the correlation mixer the energy is spread from a bandwidth of $\frac{1}{T_{chirp}}$ to the bandwidth of the reference LFM waveform. The ratio of input bandwidth to output bandwidth of the interfering tone is $\frac{1}{T_r B_r}$ where T_r is the pulsewidth of the reference waveform and B_r is the bandwidth. The ratio of the output

signal to interference to the input signal to interference for single interference is

$$\frac{SNI_{out}}{SNI_{in}} = T_{chirp} T_r B_r \quad (3.72)$$

$$= G_p B_s T_r B_r \quad (3.73)$$

More generally, the signal to interference gain for stretch processing can be described as the processing gain multiplied with the ratio of the output of bandwidth of the interfering signal divided by the input bandwidth of the interfering signal.

4.1 Group Delay Compensation

Nonlinear phase response in the receiver adversely affects stretch processing by smearing the received signal across a wider bandwidth than expected. One of the biggest sources of phase nonlinearity is at the edge of the passband in an amplifier. One solution to this kind of phase nonlinearity is to slightly extend the passband. There are two possible negative affects to increasing the bandwidth. First, the gain of the amplifier may be reduced. Second, increasing the bandwidth will also increase the noise bandwidth. As a concrete example of group delay, Figure 4.1 is a fifth order Butterworth LPF. The GD slope gets steeper as it nears the cutoff frequency. Remember group delay variation, not constant group delay, corresponds to phase nonlinearity.

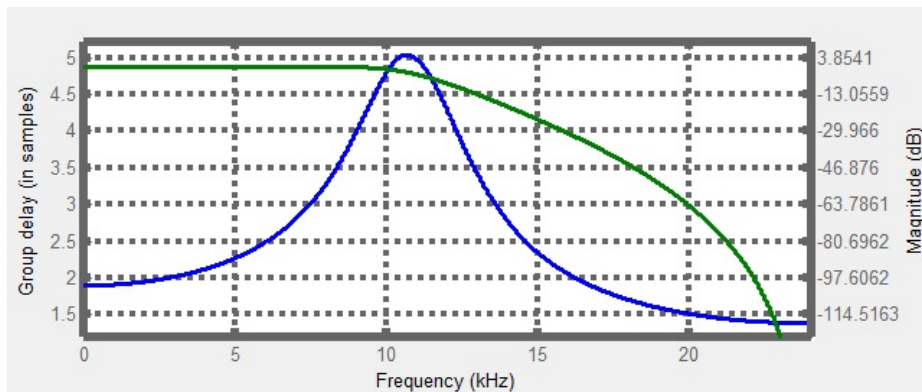


Figure 4.1: Fifth order Butterworth LPF with GD response

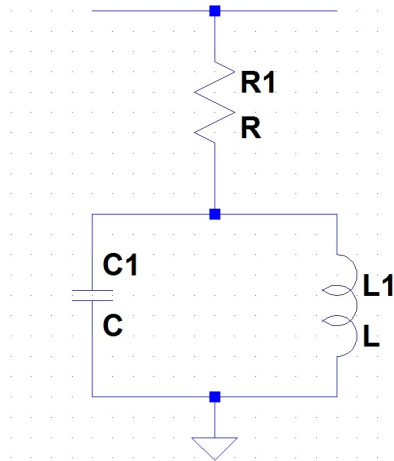


Figure 4.2: Equalizer

4.1.1 Equalizer

If the only circuit parameter of concern is group delay, maintaining constant group delay would be easy. For example, a simple notch filter circuit has a positive phase and therefore a negative group delay. Using a notch filter to compensate for group delay would filter the frequencies that need to be corrected! Some filter types have better group delay responses than others. The Bessel-Thomson filter has an optimally flat group delay response. This filter generally has unacceptable stop-band attenuation and a wide transition band.

The group delay variation occurs when the amplitude of an amplifier is starting to fall off. A circuit with a positive slope of equal magnitude where the amplifier magnitude starts decreasing will smooth the group delay. A circuit that meets this description can be described as an equalizer [11]. It's called an equalizer because it flattens or "equalizes" the gain. A common equalizer is based on the pi attenuator. This design does not work well for ICs though. Passives, especially inductors, take up too much space to be used without discretion. A simpler equalizer is given in Figure 4.2. This equalizer only requires one inductor.

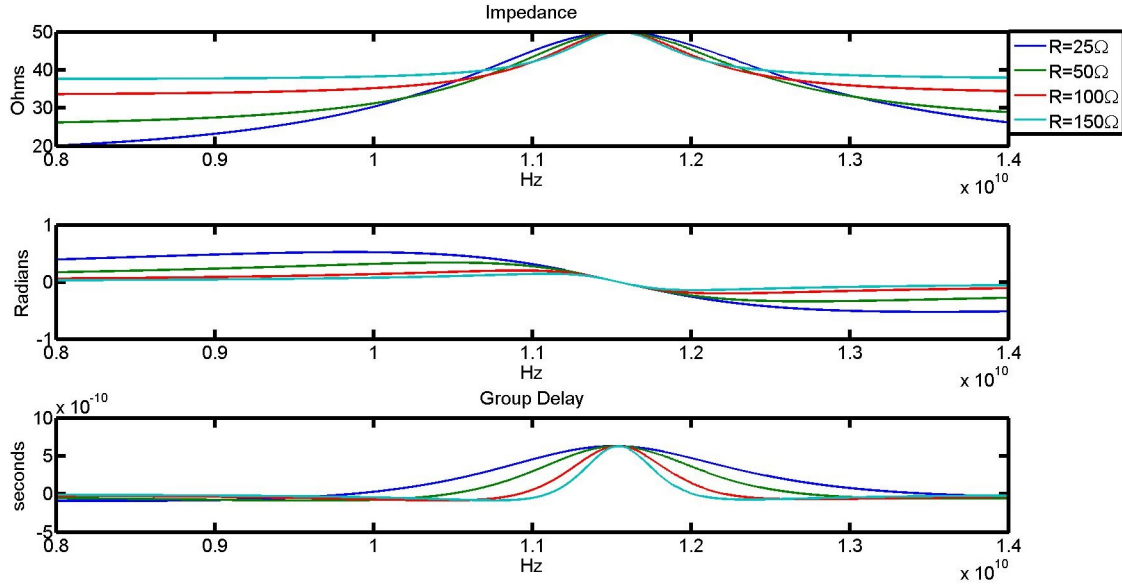


Figure 4.3: Equalizer plots a) impedance b) phase c) group delay

The impedance of the equalizer circuit is

$$y = R + \frac{j\omega L}{1 - \omega LC} \quad (4.1)$$

Assuming that it is in parallel with a 50Ω load, Z_L , the impedance is

$$Z = \frac{Z_L Y}{Z_L + Y} \quad (4.2)$$

The values of the components are chosen to meet system specifications. L and C are chosen to adjust the resonant frequency and Q-factor. While maintaining the same resonant frequency higher inductor values give lower Q-factors. Higher Q-factors provide a steeper slope; lower Q-factors provide wider bandwidth. Lower resistance values give a less steep and more linear slope than high resistance values. The resistance values are limited by the return loss of the equalizer though. Away from resonance, the the impedance of the circuit approaches R for low values. This decreases the matching of the circuit. Figure 4.3 shows plots of the impedance, phase and GD of the circuit. The range of R values is from 25Ω to 150Ω .

One reason that the circuit is so convenient is that it can be inserted into existing designs. This may be easier than re-designing an amplifier to meet group delay specifications. Also, standard amplifier design processes can be followed, and the equalizer can be added at the end if it is needed.

Before an application is given, a summary is useful. Wide bandwidth signal can suffer from phase nonlinearity. The phase nonlinearity can be characterized by the group delay which is the negative rate of change of the phase versus frequency. Constant group delay is good and represents linear phase. Varying group delay represents phase distortion. Stretch processing can be vulnerable to varying group delay. The group delay shows up in the stretch processed spectrum as spreading the bandwidth of the single-tone frequency. Phase nonlinearity is usually the worst at pass band edges of amplifiers and filters. An equalizer can be inserted into a design to alleviate the group delay variations at a pass band edge.

Note It could be a good idea to design a DDS to pre-compensate for both group delay and amplitude variations. Correcting issues digitally may be preferable to changing a hardware design.

4.1.2 LNA with Equalizer

Narrowband LNA design techniques for SiGe technology are widely available. Particularly, the design techniques presented in [5] result in well matched, high performance LNAs. As the previous references suggest, the cascode is a good configuration for LNA design. The cascode configuration is a common-emitter amplifier with a common-base amplifier connected to the output. This design has several advantages:

1. The Miller capacitance of the common-emitter amplifier is reduced by connecting it to the common-base amplifier. This increases the high frequency performance.
2. The isolation S_{12} between the input and the output is increased relative to just a common emitter amplifier.

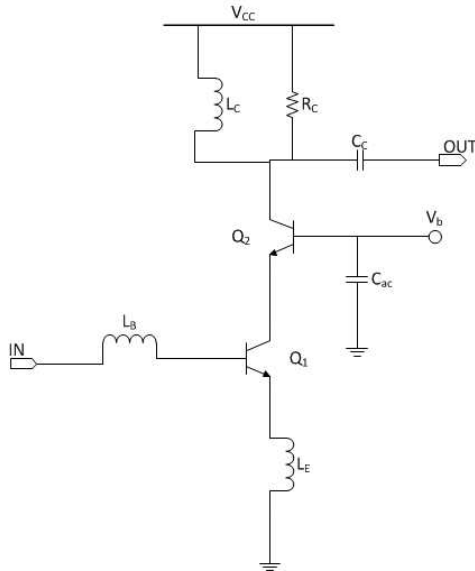


Figure 4.4: Cascode LNA

3. Because of the high frequency performance of the cascode, there is a plethora of literature discussing design techniques.

A disadvantage of the cascode is that it requires a higher voltage because of the two transistors are stacked. A simplified schematic of the cascode is shown in Figure 4.4. The design process discussed in [5] is loosely followed in the following steps to determine the parameters of the LNA. With any LNA design power consumption, noise figure, gain, linearity, stability, and maybe the bandwidth have to meet certain specification. Usually, there are trade offs for increasing any one performance metric. For example, reducing power consumption may reduce gain and linearity.

1. The first step is to determine the optimum current density for the cascode. This can be done in simulation by setting the emitter lengths at a mid-range value and sweeping the collector current. Find the collector current that gives the lowest NF_{min} and calculate the current density at this collector current. Since the optimum noise figure is not very sensitive to I_c changes, it may be increased from the optimum value to increase gain and linearity. The current density is the ratio I_c to emitter area.

2. A NF_{min} has an optimum source resistance associated with it. This is the value that the source resistance needs to be matched to for achieving the minimum noise figure. Assuming that the LNA is designed for a 50Ω system, it is desirable for the optimum source resistance $R_{s,opt}$ to also be 50Ω . The emitter length can be scaled while maintaining the same current density to set $R_{s,opt}$. This will increase the power consumption. If power consumption is a priority, choose I_c first. Then select emitter length to balance noise figure and gain.
3. The next step is to set L_B and L_E . L_E provides inductive degeneration and matches the real part of the input impedance. Inductive degeneration increases linearity but decreases gain and increases noise figure. The equation for the emitter inductor is

$$L_E = \frac{50}{2\pi f_T} \quad (4.3)$$

where f_T is the unity gain frequency for the particular current density. The base inductor cancels the base-emitter capacitance. This capacitance increases as the emitter length increases. Correctly setting L_B will match noise and the input. The equation for the base inductor is

$$L_B = \frac{1}{\omega^2 C_{BE}} - L_E \quad (4.4)$$

These equations are approximations. It is recommended to use the values as starting points and sweep sets of inductor values in simulation to find the best values.

4. All that is left is the output matching. R_C is used to decrease the Q-factor and increase stability. It may not be needed. The LC network can be determined using a Smith chart [13].

The preceding design steps are for a narrowband design. Recall that group delay is also a concern for this LNA. Group delay is worst around band edges. As an example, an X-band LNA is designed. After it is designed, an equalizer is added to its output. First look at

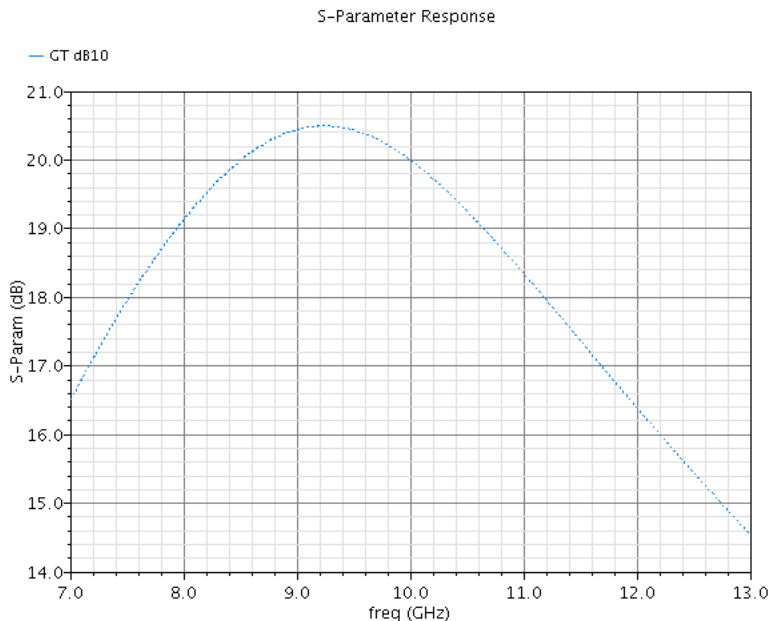


Figure 4.5: LNA gain with no equalizer

the gain before and after the equalizer is added. Figure 4.6 shows that the LNA with the equalizer has a much flatter gain. Without the equalizer the gain drops from 20.4 dB at 9 GHz to 18.3 dB at 11 GHz. The equalizer flattens the gain. It varies only 0.8 dB from 9-11 GHz. The peak gain is dropped to 18.6 dB.

It is important to note that the noise figure is unaffected by the equalizer. The noise figure for both circuits is shown in Figure 4.7. As expected, the group delay decreases quickly at the band edge. The group delay changes by about 40 picoseconds in the circuit with no equalizer. The equalizer group delay drops by 15 picoseconds between 9 and 10 GHz before it starts increasing to the starting level at 9 GHz.

There are two major drawbacks to using the equalizer. First is the added space that the circuit takes up. The inductor is especially large. Second the output matching is decreased. The value of the equalizer resistance controls the output matching. The value of the resistor in the equalizer was selected to keep S_{22} below -10 dB for the whole band of interest. Figure 4.8 demonstrates how the equalizer changes the circuit from narrowband output matching to worse but wideband output matching.

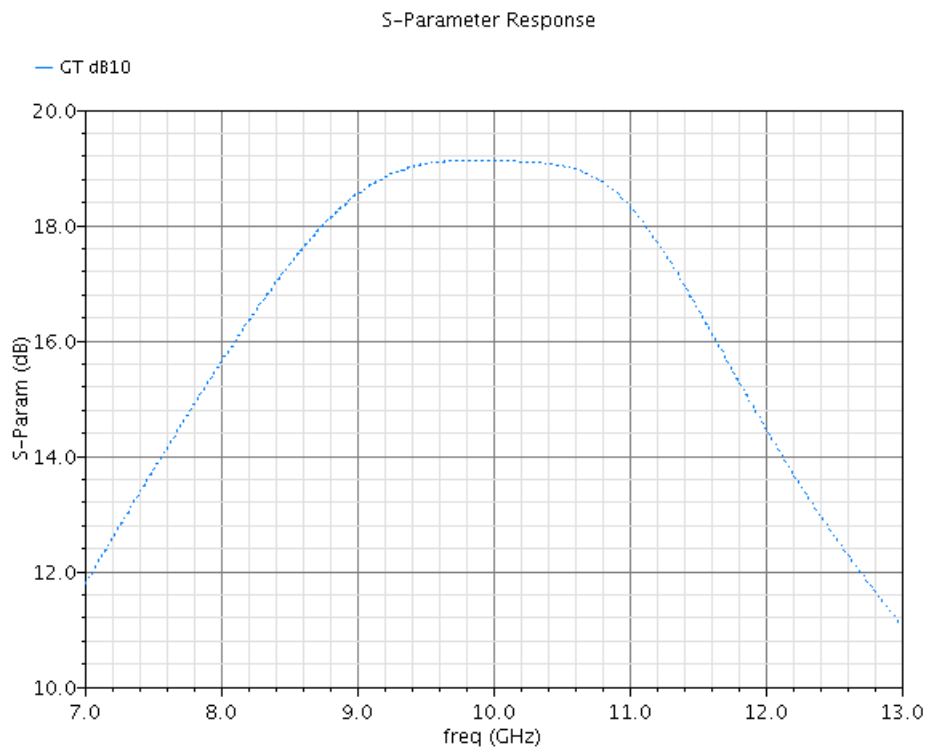


Figure 4.6: LNA gain with equalizer

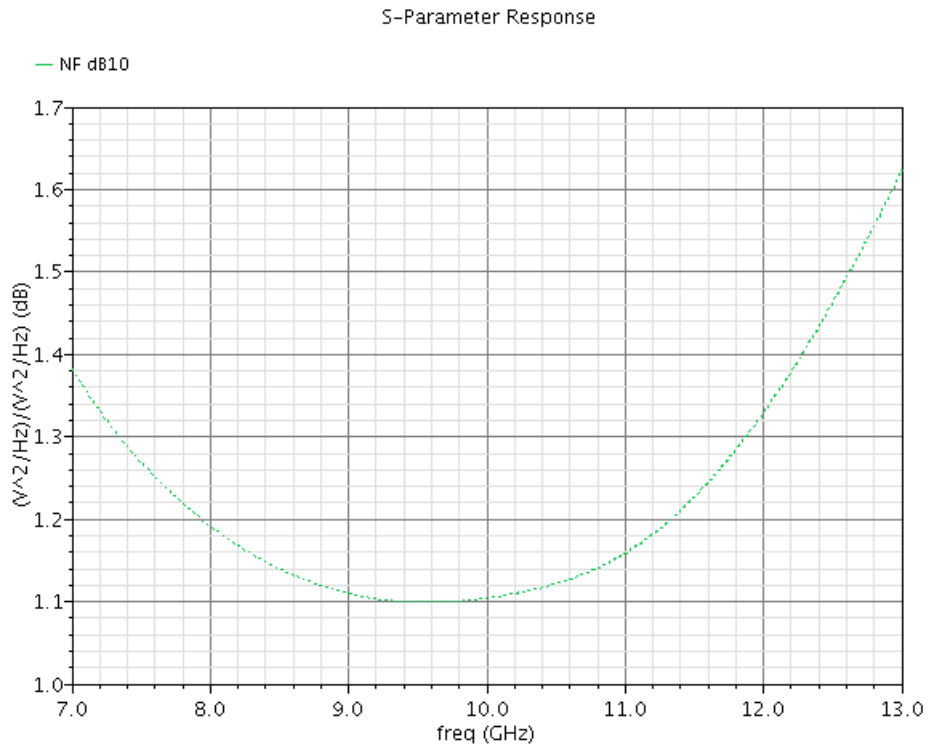


Figure 4.7: LNA NF

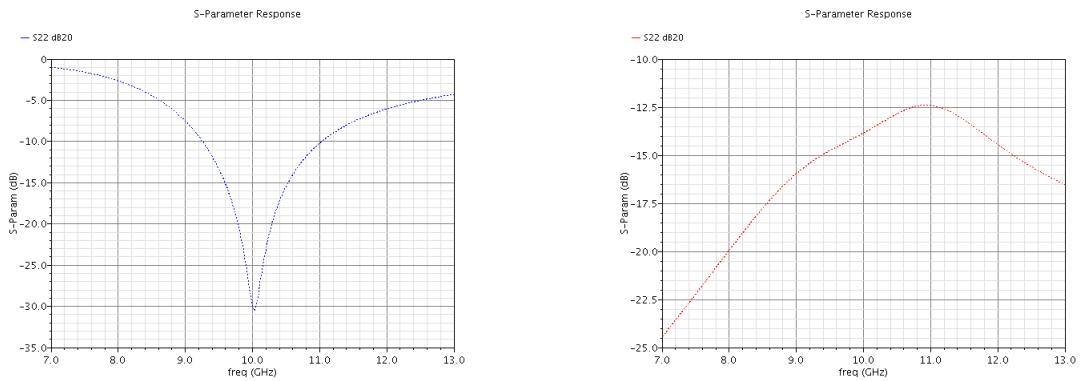


Figure 4.8: a) S_{22} with no equalizer b) S_{22} with equalizer

Also, the resonant frequency of the equalizer can be moved lower to decrease both the bandwidth and the group delay variation. There is no need to have a wider bandwidth than needed. That will only add to the noise. In conclusion, a cascode is arguably the best configuration for high frequency SiGe LNAs. Due to their widespread use, optimized design processes are available. Adding an equalizer to a circuit allows for the use of narrowband designs while still taking into account bandwidth and group delay.

4.2 Common-Base Amplifier

The common-base (CB) amplifier also has good high frequency performance because there is no Miller effect. The input impedance for the CB amplifier is approximately $\frac{1}{g_m}$. This can be much less than the typical 50Ω system and creates a challenge for achieving a good input match. The voltage-voltage feedback solves the problem of low input impedance. Figure 4.9 shows a simplified schematic of a CB amplifier with voltage-voltage feedback. The idea of using voltage-voltage feedback to control input matching for a high frequency SiGe CB LNA is presented in [6]. The voltage gain of a CB amplifier is approximately $g_m R_L$. The current gain is approximately 1. In a 50Ω system, a gain of $50g_m$ may be too low and current gain may be needed to drive the next stage. The most obvious solution to this issue is to follow the CB amplifier with a common-collector (CC) amplifier. This resolves several issues. First, the current gain of a CC amplifier is $\beta_0 + 1$. Also, the input impedance of a CC amplifier is

$$Z_{in} = r_\pi + (\beta_0 + 1)Z_L \quad (4.5)$$

The load impedance is magnified by $\beta_0 + 1$. Since this is the impedance that the CB stage sees, a much larger gain is achieved than by just a CB stage with a 50Ω load.

After deciding on a circuit configuration, the next steps in designing an LNA are setting the bias current and sizing the transistors. The first step is to choose a current density J_c

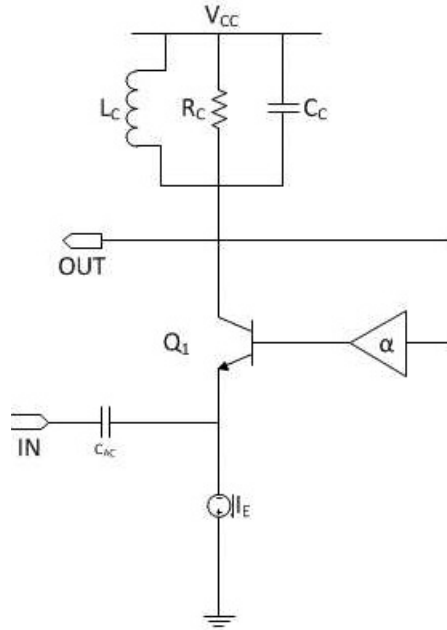


Figure 4.9: Common-base amplifier with voltage-voltage feedback

that gives a low minimum noise figure for the CB amplifier. To determine the optimum J_c , I_c is swept for a fixed emitter length. Figure 4.10 shows the results of this sweep. The minimum noise figure is somewhere between 4 mA and 5 mA. 4.5 mA is chosen which corresponds to a J_c of $2.5 \text{ mA}/\mu\text{m}^2$.

The emitter length is determined next. Unlike the cascode, varying the emitter length is not used to set the input impedance. This allows for more freedom in setting the emitter length. Figure 4.11 is a graph of the minimum noise figure (NFmin) versus emitter length. The NFmin drops slightly as the emitter length is increased. A much smaller emitter length can be used compared to a cascode LNA at the same frequency, since the base resistance is not as important. This allows for a transistor with a high f_T to be chosen. An emitter length of $20 \mu\text{m}$ with a corresponding I_c of 6 mA is chosen to compromise between noise figure, power consumption, and linearity.

As already described, the amplifier is a two-stage CB-CC amplifier. The implementation uses a current sharing scheme to save power. The CC amplifier is stacked on top of the CB

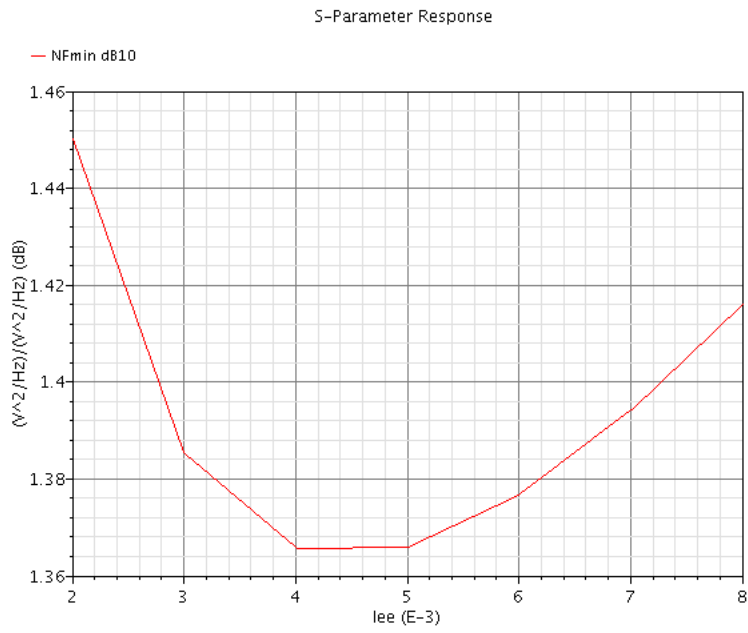


Figure 4.10: NF_{min} vs. I_{EE} for determining J_c

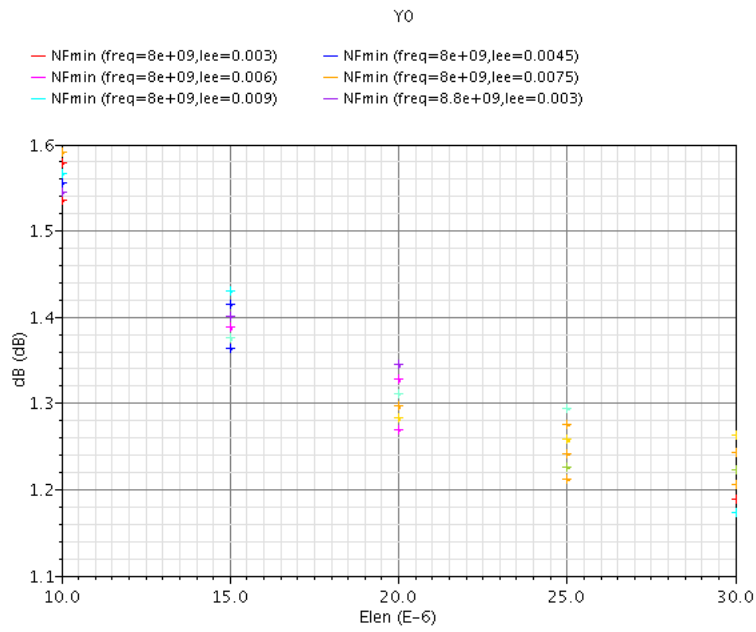


Figure 4.11: NF_{min} vs. emitter length

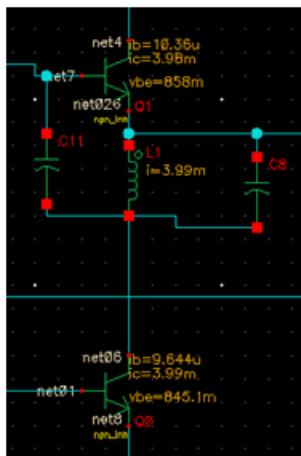


Figure 4.12: The interface between CB and CC

amplifier. There is an LC tank circuit connected to the collector of the CB and the emitter of the CC. This tank circuit allows the DC current to pass through the inductor, but represents high impedance to the frequency band of interest. Figure 4.12 shows a zoomed in picture of the connection. The capacitor C11 in Figure 4.12 represents an ac short; thereby, letting the output signal of the CB amplifier pass to the input of the CC amplifier. Although this may look like some kind of oscillator, the amplifier is unconditionally stable in the frequency band of interest. The emitter length of the CC amplifier is $15 \mu\text{m}$. It is smaller than the CB amplifier to increase the input impedance, β_0 , and linearity of the CC amplifier.

The calculation of the input impedance can be done using feedback amplifier theory.

$$Z_{in} = Z_{in}^{\text{open loop}}(1 + \alpha G_{\text{open loop}}) \quad (4.6)$$

For this circuit, Equation 4.6 reduces to

$$Z_{in} = \frac{1}{g_m} + \alpha\beta_0 Z_L \quad (4.7)$$

The β_0 is present because of the magnification of the load impedance by the CC amplifier. The input impedance is set by the output impedance. Because Z_L is a function of frequency,

the input impedance can be tuned by tuning the output load. Ideally if α is correctly chosen, the input and output impedance can be simultaneously matched for a range of frequencies by changing a single reactive element in Z_L . For implementing the design, the output is matched with a lumped element matching network. The output matching is done before adding any feedback. A Smith chart is used to design the matching network by moving the output load to S_{22}^* .

Next, the feedback is added. Figure 4.13 shows the schematic of the whole amplifier. The biasing circuitry is not shown. The feedback network is formed by the capacitive divider consisting of C25 and C17. $\alpha = \frac{C_{25}}{C_{25}+C_{17}}$. C25 is chosen as small as possible so that it will be high impedance to the operating frequency. This is needed to keep it from affecting the output matching network. For this reason, C25 is set at 100 fF which is the lower limits of the technology. Unfortunately, there is still a noticeable impact on the output matching. In the current implementation, C17 = 1.1 pF. This results in $\alpha = 0.08$. This feedback factor is small enough that the gain is not affected very much, but it is the required factor to match the input. The two capacitors are also theoretically noiseless elements, so the noise figure should not be affected by them.

The supply voltage is 2.2 V and the current consumption is 6 mA resulting in a power consumption of 13.2 mW. One issue to consider is that when the bias circuitry is inserted, the ideal current source will be gone. The signal injection still needs to see high impedance here. This could pose an issue for an on-chip inductor. A similar implementation has used a high-quality surface mount inductor off chip. An improvement is the elimination of a large off-chip inductor at the signal injection. A tank circuit with a much smaller inductor realizes the high impedance that needs to be seen by the input signal.

4.2.1 CB-CC Simulation Results

The simulation was done in Cadence IC Design 6.1. S-parameter, periodic steady state (for linearity analysis), and dc simulation were required to get all the results. Figure 4.14

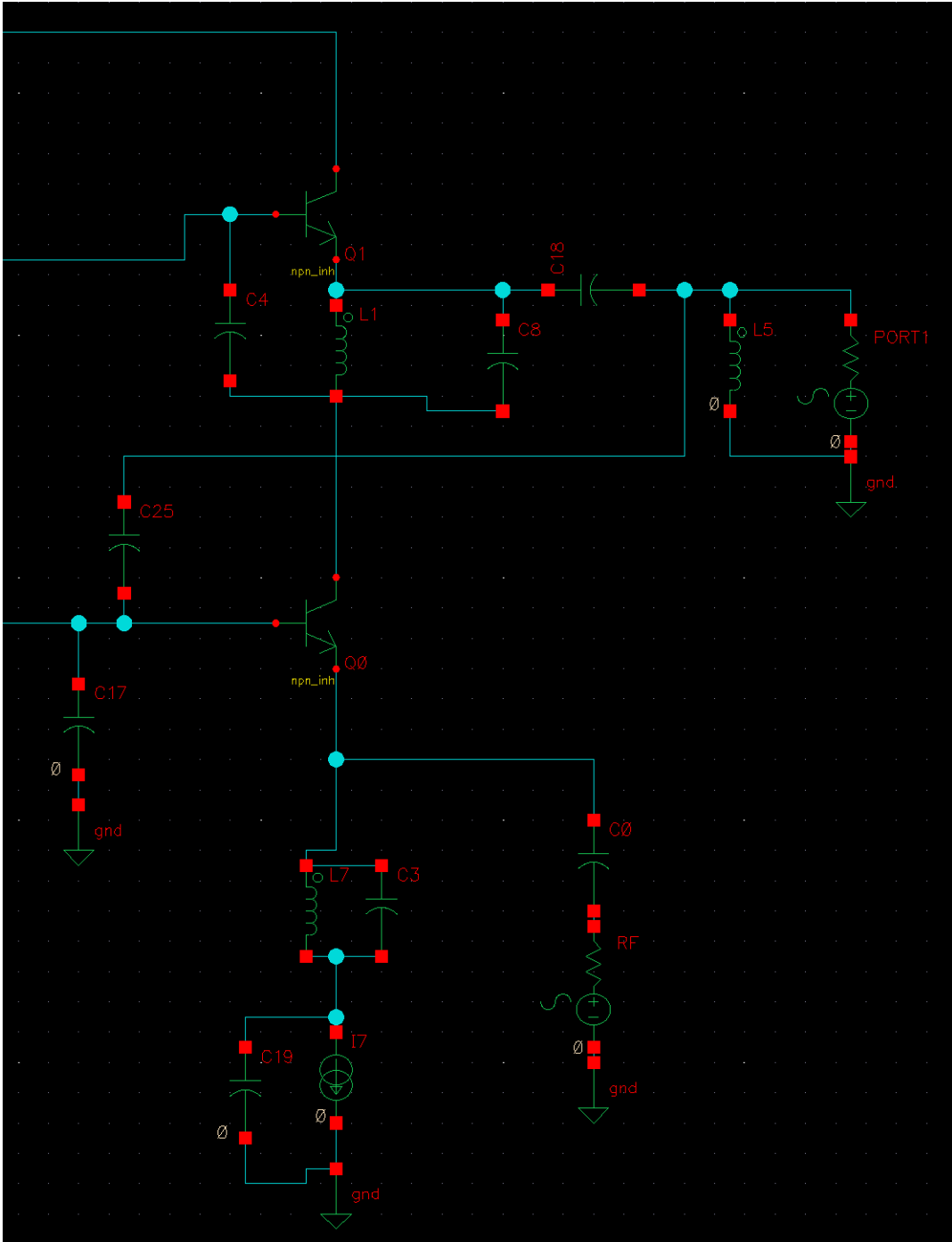


Figure 4.13: Schematic of CB-CC LNA

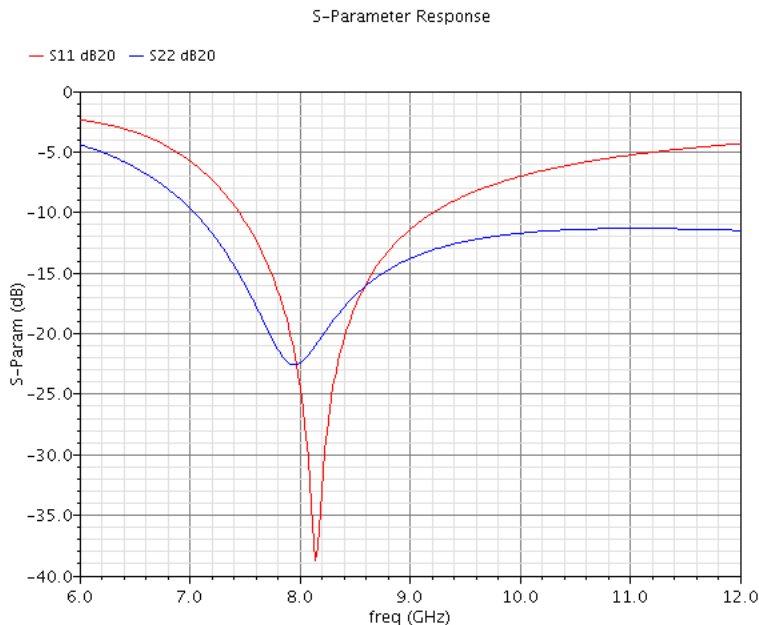


Figure 4.14: S_{11} and S_{22} are simultaneously matched

shows the simultaneously matched input S_{11} and S_{22} . Recall that this is with no standard input matching network, only the capacitive divider feedback. Figure 4.15 is the real part of the input impedance. It reaches 50Ω right at 8 GHz. Without the feedback, the input impedance would be much less than 20Ω . A possible use for this LNA circuit is for use in non-fifty Ohm systems since the input impedance is easily set. The noise figure is 1.7 dB at 8 GHz. This is with ideal passive components, but it should remain reasonable when $0.13 \mu\text{m}$ technology library parts are exchanged for ideal parts. Figure 4.16 is the NF and the NFmin. There is only 0.5 dB difference between the NF and NFmin.

Figure 4.17 is the gain of the LNA. Although the peak of the gain is not exactly where S_{11} and S_{22} are matched, there is plenty of gain at 8 GHz. The gain variation at 8 GHz is not significant for narrow bandwidths. The gain at 8 GHz is 18.6 dB.

The amplifier simulated above has a novel architecture and excels in some performance areas. The gain, matching, power consumption, and stability are all exceptional. This amplifier may be a viable alternative to the cascode structure. Setting S_{11} only by using

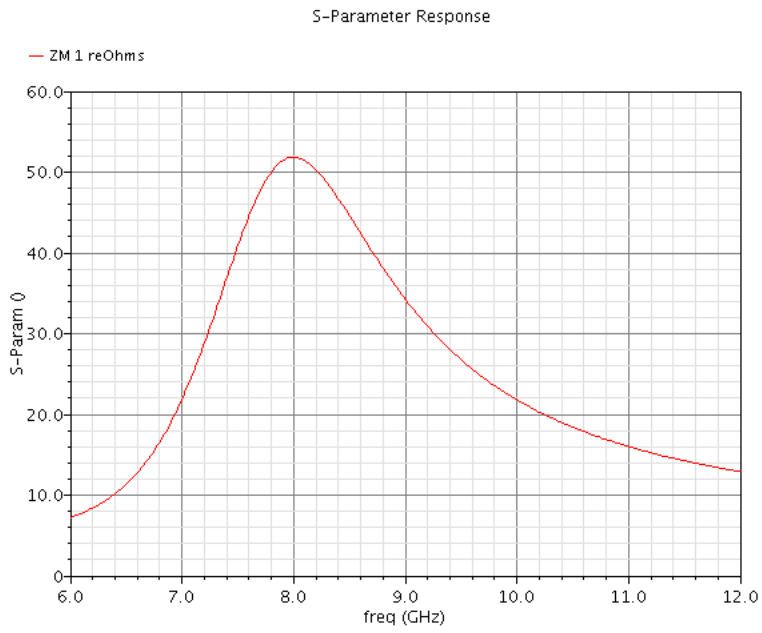


Figure 4.15: Input impedance of CB-CC

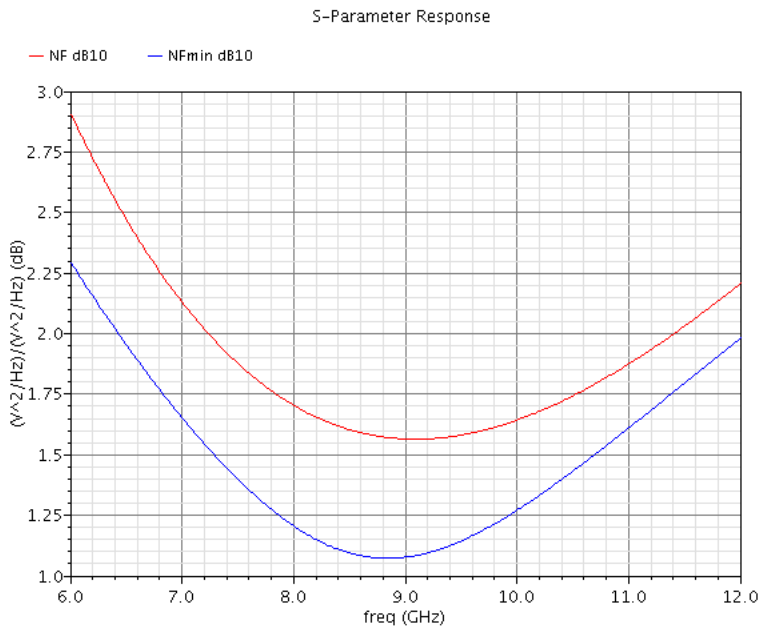


Figure 4.16: NF and NFmin comparison

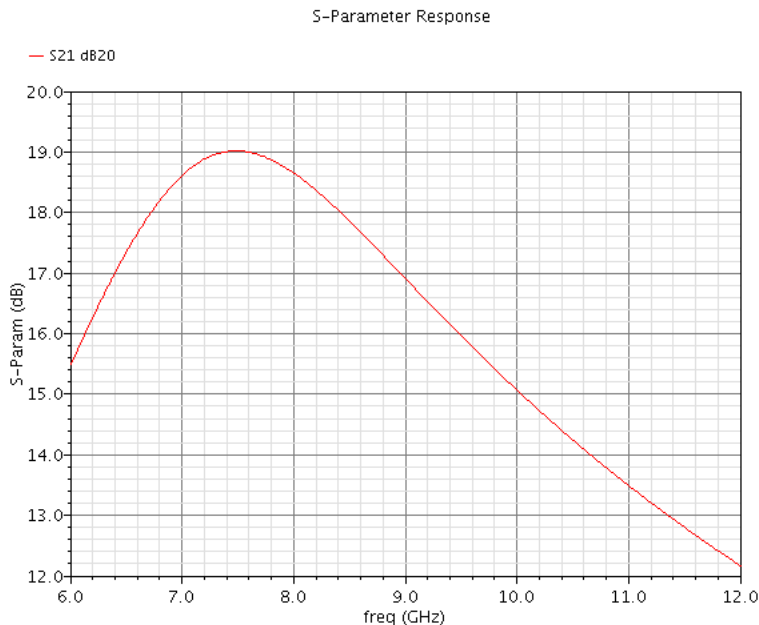


Figure 4.17: CB-CC Gain

the output load is especially desirable. It requires three inductors which is the same as the standard cascode structure. 4.1 is a performance summary of the LNA.

Table 4.1: CB-CC performance summary

Frequency	8 GHz
Gain	18.6 dB
Power	13.2 mW
NF	1.7 dB
S11	-25 dB
S22	-23 dB
P1dB	-19 dBm
IIP3	-9 dBm

4.2.2 Integrated Duplexer

The authors of [8] and [9] have suggested the possibility of integrating a duplexer using a hybrid transformer [10] and electrical balance. A hybrid transformer has several useful characteristics:

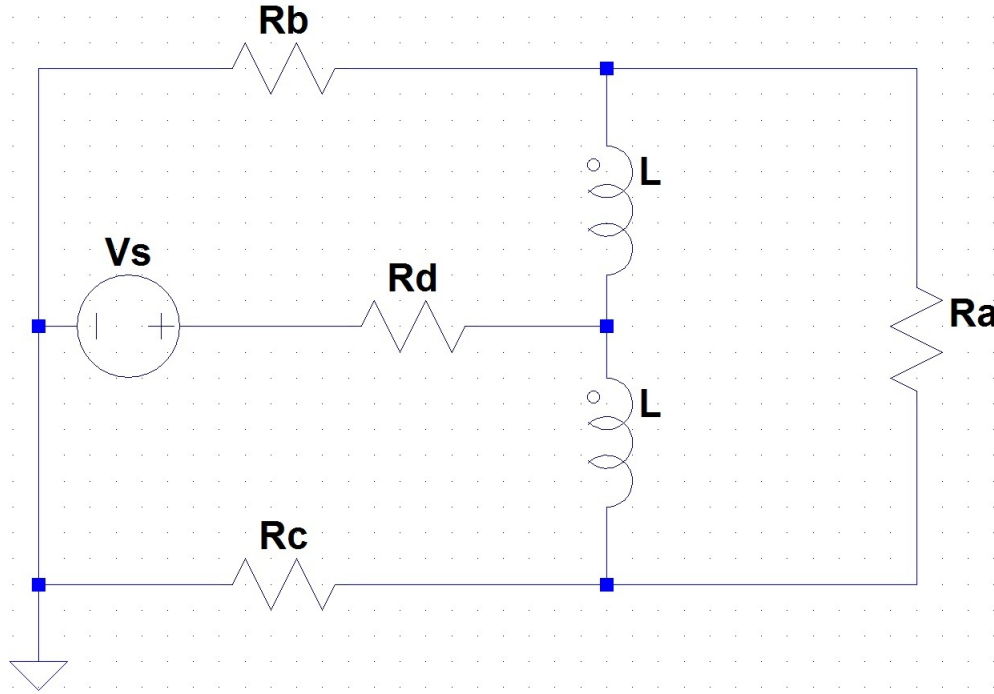


Figure 4.18: Autotransformer used as hybrid transformer

- All ports can be simultaneously matched.
- The ports are bi-conjugate. This means that TX and RX can be electrically isolated.
- The incoming power from one port can be split into any ratio of power to the receiving ports.

The duplexer is realized with an autotransformer. An autotransformer is a simple implementation of a hybrid transformer. It has only one winding and a center tap. The autotransformer can be fabricated on an IC with a differential inductor. This simple hybrid transformer is shown in Figure 4.18. The autotransformer can be matched with the following set of equations:

$$R_c = R_b$$

$$R_a = 2R_b$$

$$R_d = R_b/2$$

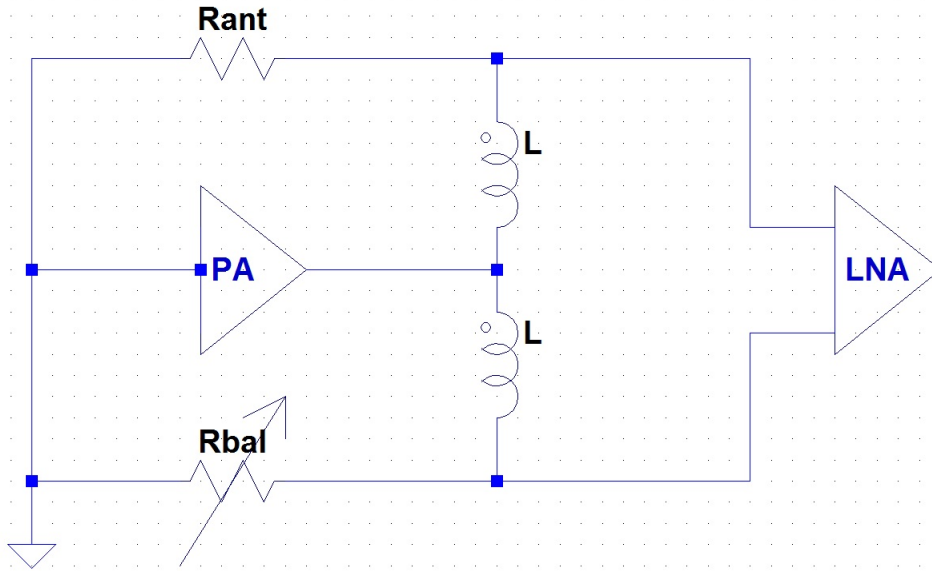


Figure 4.19: Duplexer

Some port terminations in the duplexer cannot be 50Ω . If the ports are matched, R_a and R_d are isolated; and R_b and R_c are isolated. [2] selected the configuration in Figure 4.19 as the optimum configuration of the autotransformer. The power amplifier cannot be integrated, so it is off-chip. A major concern for the duplexer is that it cannot handle the power from the power amplifier. The TX to RX isolation is very good. It is also very wide bandwidth. The bandwidth of the isolation is limited by parasitics. R_{bal} in Figure 4.19 is a tunable resistor. The power amplifier output appears as common-mode voltage to the LNA. The LNA and the power amplifier do not have 50Ω matched states. This is why the CB-CC amplifier already described in this section is viable for the LNA.

The LNA has to be modified to be differential. The input impedance should be 100Ω if the antenna impedance is 50Ω . The duplexer and LNA were simulated with modest results. The S-parameters are in Figure 4.20. S_{22} is the only concern with slightly more than -10 dB in the frequency band.

The insertion loss for the PA is about 5 dB. A higher tuning resolution should reduce this. The isolation from the PA to the LNA is about -50 dB and very wideband. The reverse

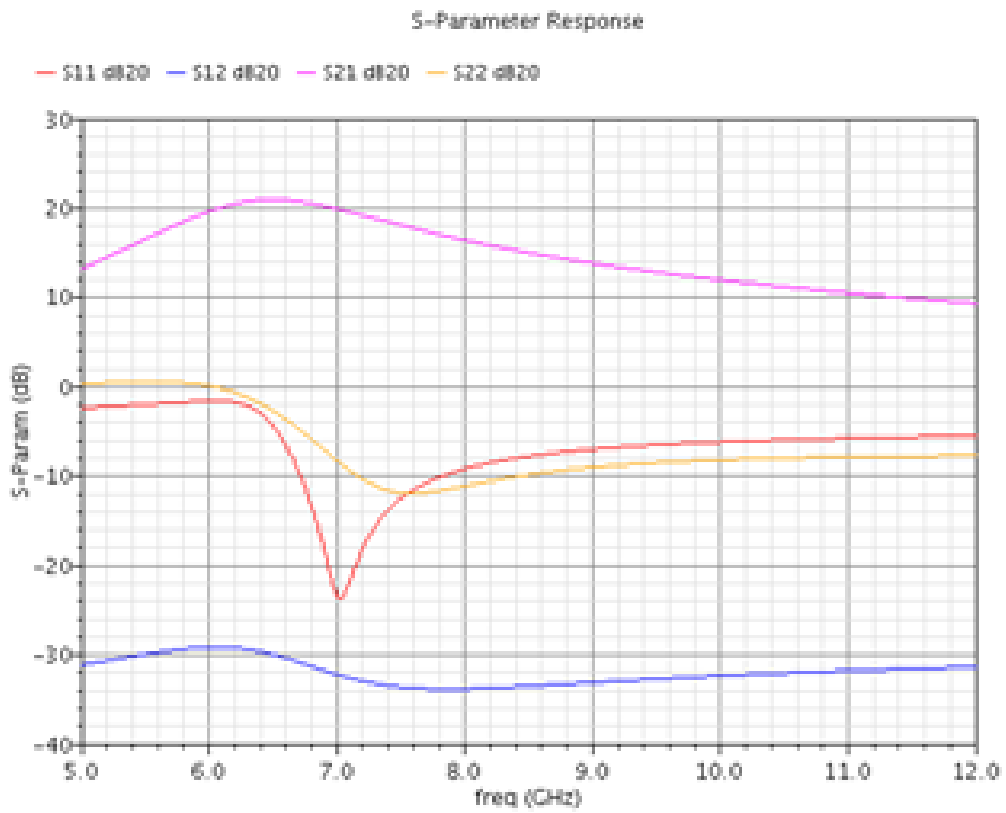


Figure 4.20: S-parameters for LNA with duplexer

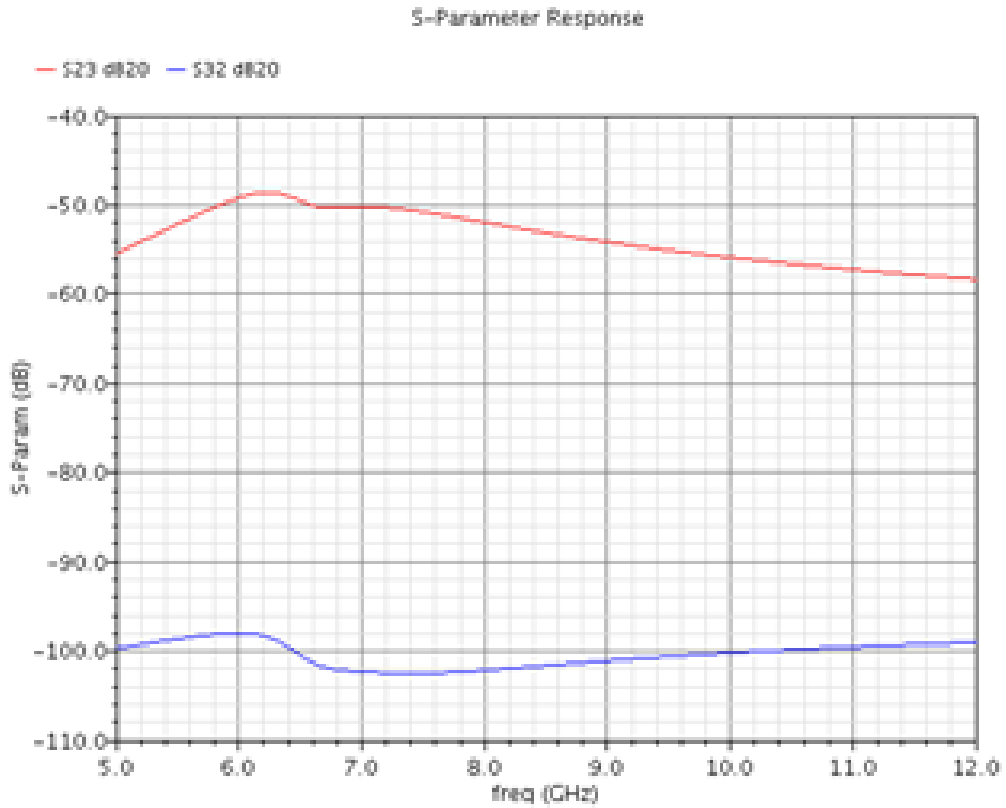


Figure 4.21: Duplexer isolation

isolation from the LNA output to the PA is greater than 100 dB. Figure 4.21 is the isolation. The noise figure is made much worse by adding the duplexer. The insertion loss of the RX adds to the noise. It is slightly less than 5 dB as shown in Figure 4.22.

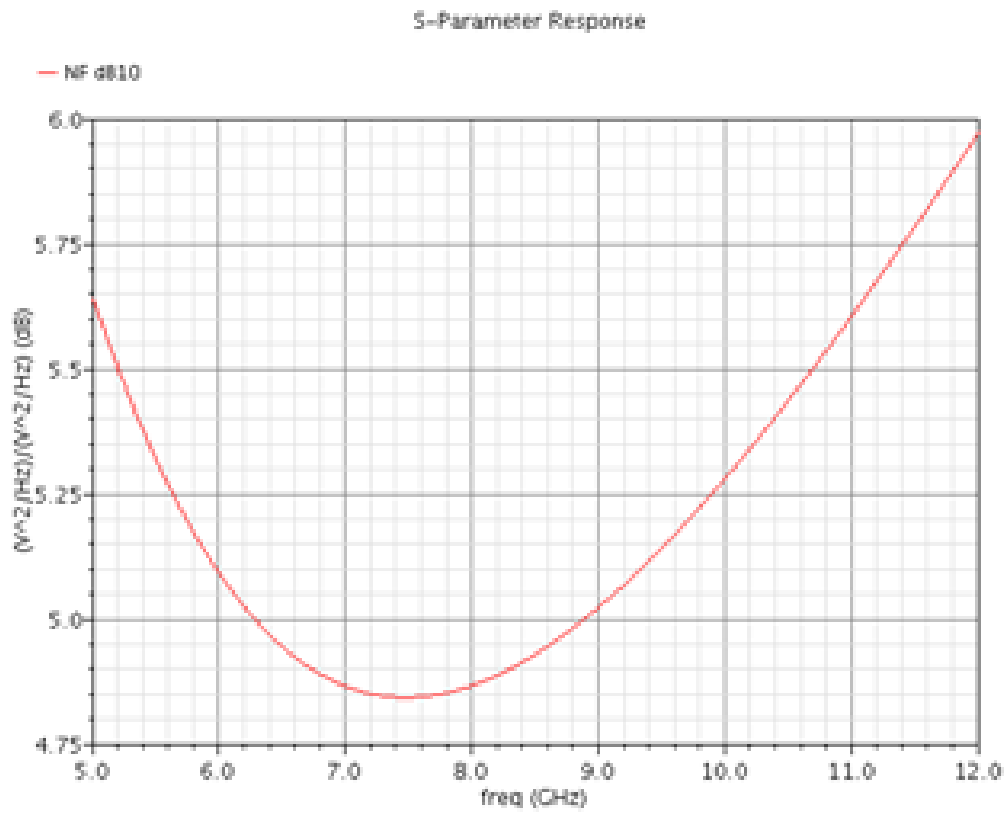


Figure 4.22: Duplexer and LNA NF

Chapter 5

Software

5.1 Test Setup

The ROC package has about 100 pins. Since the ROC had to be carefully tested, dozens of testing pins were used to monitor the system at different points during transmitting and receiving. Over thirty pins are used for VDD, GND, and voltage references. There are also pins for clocks, connecting filters to the IC, and the ADC outputs. The point is that pin real estate on the IC is scarce. This makes serial communication necessary for controlling the dozens of control bits for the ROC. A serial peripheral interface bus or SPI was designed to control the ROC.

A SPI has a master and a slave. It only requires four lines for communication:

SPI_DI: serial data input

SPI_DO: serial data output

SPI_CLK serial clock

SPI_CS chip select

Figure 5.1 is a simple diagram of how the spi communication is wired. The ROC is controlled with a Xilinx Spartan-6 FPGA on an SP601 evaluation board. The SP601 is the master

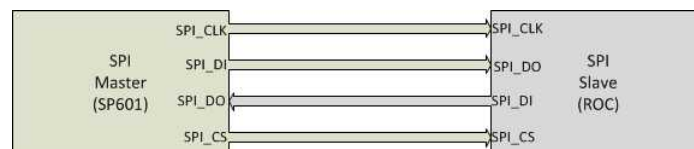


Figure 5.1: SPI diagram

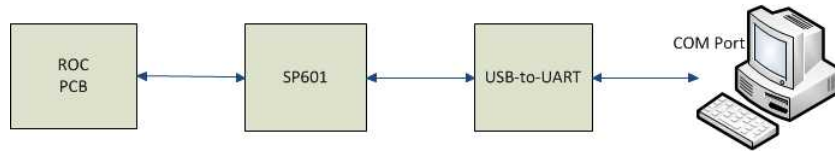


Figure 5.2: ROC communication diagram

device in the SPI. The ROC is the slave. The ROC itself is on custom designed PCB for testing. The SP601 communicates with a computer through a USB-to-UART device. The USB-to-UART device converts serial data transmission from the Spartan-6 to comply with USB transmission. This allows the SP601 to communicate with a computer through a standard USB connection. UART stands for universal asynchronous receiver/transmitter. It translates the parallel data form internal to the Spartan-6 to serial form. The pins for UART transmission are provided on the Spartan-6 and connected to the USB-to-UART chip on the SP601. The UART itself has to be custom designed by the user.

The USB-to-UART device manufacturer provides a free Virtual COM Port device driver which allows the USB-to-UART device to appear as a COM port on a computer. The block diagram for how the ROC communicates with a computer during testing is shown in Figure 5.2.

The tester needs an intuitive, quick way to communicate with the ROC. Designing good software will save time and reduce errors in testing.

5.2 Software Needs

To begin with, the features and functionality of the software need to be defined.

- The most important function of the software is to send updates to the ROC. This most basic functionality could be provided with communication software such as HyperTerm. This would require sending a string of ascii characters such as “asdf.” There is no information about what is being sent to the ROC. Ideally, information about the ROC control bits is available to the user. The ROC control bits are divided into different

registers such as the PLL register, ADC register, or DDS register. The software should be able to select a register or individual bits of a register to update. The current state of each register should also be available.

- With hundreds of different control bits in a dozen registers, configuring the ROC to a particular state can be cumbersome. Saving and loading ROC states will save time and enable consistent test setups. The saved format of test states should also allow editing or the creation of new test states.
- Modular software design permits the software to be dynamic and easily updated. New needs may arise during testing. Communication protocol with the ROC may change if the SP601 is changed or replaced. Changing as little of the software as possible as conditions change ensures that testing time will not be lost.
- The software needs to be platform independent. For obvious reasons, the software does not need to be tied to any one computer.
- The software has to be able to read and store information from the ADC. Viewing the time and frequency data from ADC reads is another good feature.
- The user interface has to be intuitive and easy to learn. One option is a command line interface. A command line interface is inherently not as easy to use as a well-designed GUI. A command line interface requires a user to navigate tiers of menus for each action. A GUI, while taking more development time, is a better option for ease of use.

5.3 Platform Choice

Python is the language chosen for software development. The Python v2.7 help documentation describes Python as follows [12]:

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented

programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and extensive library are freely available in source or binary form for all major platforms from the Python website... and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

What is not to like about Python!? One key thing about Python is the plethora of third party packages that address the needs of the ROC control software. Python is easy to learn. It is also free. There are third party packages that address COM port communication, GUI development, and Excel spreadsheet creation and editing. The ease of use of any programming language is highly dependent on the integrated development environment (IDE). Eclipse with the pydev extension was chosen as the IDE. Like Python, Eclipse is free. It provides real time parsing, code completion, debugging, a file explorer, a code outline, and a command line interface.

5.4 Code Structure

5.4.1 GUI

Python has several options for GUI development. PyQt was chosen for this project. PyQt is advantageous because it has a low learning curve before a useful GUI can be created. From PyQt's website, "PyQt is a set of Python bindings for Nokia's Qt application framework... The bindings are implemented as a set of Python modules and contain over 300 classes and over 6,000 functions and methods." The advantage of being able to use Qt for GUI development is using Qt Designer.

Qt Designer allows for the graphical design of a GUI. A GUI can be built by dragging and dropping objects such as menus, tables, or buttons onto a window. Once the GUI is

built and all of the objects are labeled intuitively, it is converted to a python class through a command line tool provided with PyQt. An instance of the class is then created in the main Python program. This process allows the GUI to be updated seamlessly as long as object names do not change.

Once the user interface is added to the main program, it has to be connected to other functions to correctly handle all events. An event is any input or action to the GUI that elicits a response i.e., a button press, a menu selected, text entered, or a tab changed. This is done using commands that connect an event to a specific function. An event can have multiple functions connected to it. A function may have multiple events connected to it.

5.4.2 ROCDriver

The core of the software is a class representing the ROC driver. This class contains all information about the ROC state, the default values, the registers, the register addresses, and functions required for updating the ROC. The ROC SPI has twelve registers. Each register has an address that is prepended to to the updates of the control bits. The registers are defined as follows:

PLL_SPI is a 64-bit register containing all of the control bits for the PLL.

ADC_SPI is a 20-bit register containing all of the control bits for the ADC.

CTRL_SPI1 is a 20-bit register controlling the overall operation of the ROC.

CTRL_SPI2 is a 64-bit register that controls the wait time after transmitting and after receiving.

DCDO_SPI1 is a 64-bit register that controls the start and stop frequency of the transmitted LFM waveform.

DCDO_SPI2 is a 64-bit register that controls the start and stop frequency of the reference LFM waveform during receiving.

DCDO_SPI3 is a 64-bit register controlling the initial phase and the frequency step of the LFM waveform.

DCDO_SPI4 is a 15-bit register controlling miscellaneous DDS functions such as DAC gain control and inverse sinc filter enable.

PA_SPI is a 3-bit register enabling the power amplifier driver and setting the power amplifier driver current.

PHASE_SPI is not implemented in the current ROC fabrication

TX_SPI is a 26-bit register that controls transmitter options such as filter tuning, VGA control, and transmitter powerdown.

RX_SPI is a 26 bit register that controls receiver options such as LNA select, mixer current controls, and VGA control.

The ROCDriver class contains all relevant information about the ROC. The necessary information is a map relating each control bit to its register, register addresses, default control bit values, current control bit values, the location of the control bits within a register, and functions necessary for communication.

One third party package is needed for the ROCDriver class. The package, `bitstring` allows for the easy storage and manipulation of arrays of bits. Each control bit can be stored separately and easily concatenated to form the register.

The GUI operations are divided into five different tabs. One tab is devoted to updating the control bits and showing the current values of the control bits. Individual control bits can be updated or entire registers can be updated. Figure 5.3 is a screenshot of the ROC interface software with the SPI tab active. The register is selected from the list on the left. The control bits are shown in the table to the right. The columns are the bit numbers, bit name, current value, the value to be written on update, and a column to enter new bit values. Once the new values are entered, the “To Be Written” button is pressed. This updates the

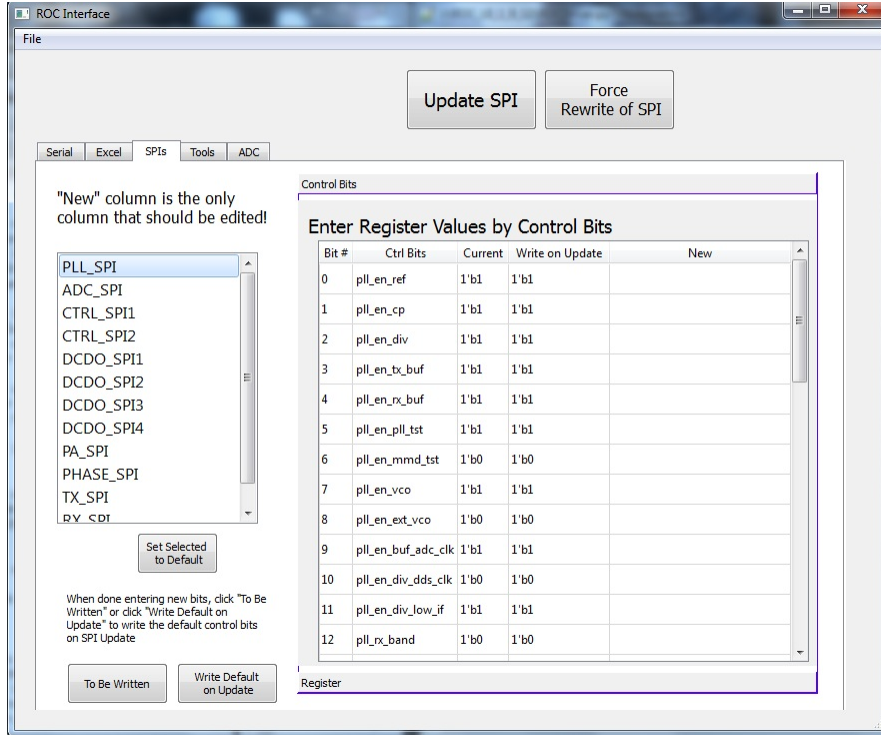


Figure 5.3: ROC Interface SPI tab

“Write on Update” column. Once all of the registers are updated, the large button at the top of the page labeled “Update SPI” is pressed to actually transmit the data to the ROC.

5.4.3 Serial Communication

The third party package `pySerial` provides access to the windows COM port. It can create an object for the COM port that can be opened, closed, read from, and written to. As long as the correct COM port is chosen, communication to the ROC is trivial. The ROC Interface software has a tab devoted to opening a COM port. It scans the computer for open COM ports first. The correct COM port is then selected and opened. A message appears stating the success or failure of the port opening. Figure 5.4 is a screenshot of the communication tab.

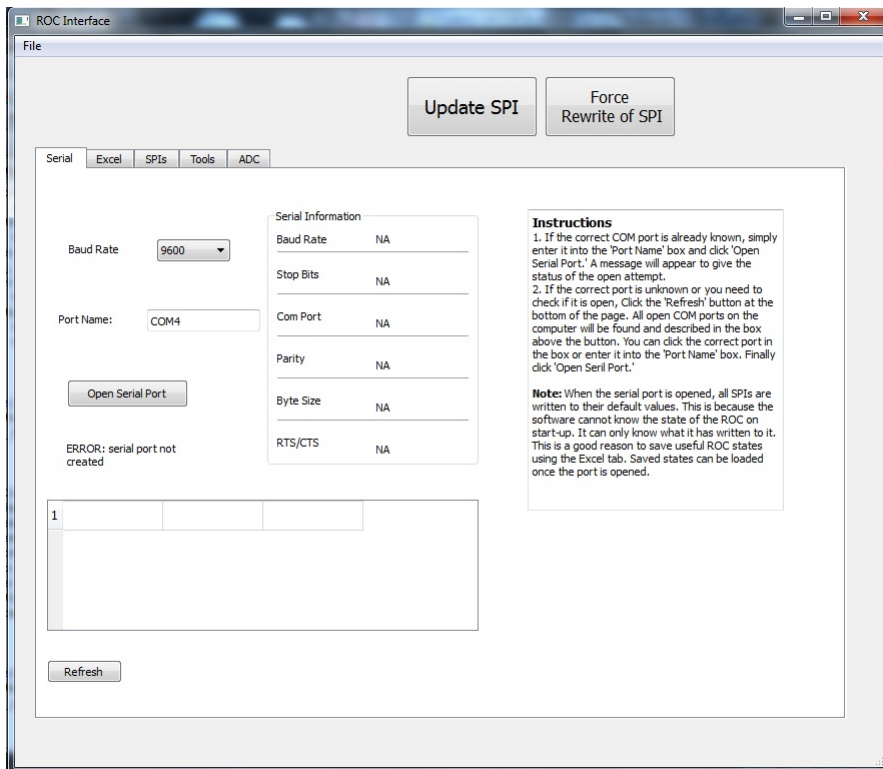


Figure 5.4: ROC Interface communication tab

5.4.4 Load and Save ROC States

Loading and saving ROC test states is very valuable to the tester. This is accomplished in the ROC Interface software with the use of Excel spreadsheets. Three companion third party packages are needed to enable the use of Excel spreadsheets. `xlrd` is used to import and read spreadsheets. `xlwt` is used to edit and save spreadsheets, and `xlutils` is used to copy spreadsheets.

When a test state is saved, all of the control bit names are put into a column sorted by registers and bit order. The bit values are placed in the column to the right. When a test state is loaded, each cell is checked to see if it matches a control bit name. If it matches a name, the cell to the right is assumed to have the values. The values are error checked to make sure it is a valid entry. This allows for test states to be created and edited in Excel. Other descriptive information can be added to the spreadsheet without causing error also.

The Excel tab of the ROC Interface is divided into a “Load Test State” section and a “Save Test State” section. The “Load Test State” section opens a workbook first. Once a workbook is loaded, the worksheet is selected. The “Load” button updates the “To Be Written” data for all of the control bits. The data is sent to the ROC once the “Update SPI” button at the top of the GUI is pressed.

The “Save Test State” can save a test state to an existing workbook or create a new workbook. Worksheets can also be overwritten. There is a field to enter a description. The description is displayed when the state is selected to load. Figure 5.5 is a screenshot of the Excel tab. The top is the load section, and the bottom is the save section.

5.4.5 Tools Tab

The ROC Interface has a tab for miscellaneous operations. For now its main function is to calculate the correct 32-bit word for a given frequency. This function is necessary for the start and stop frequencies of the transmit and receive LFM waveform.

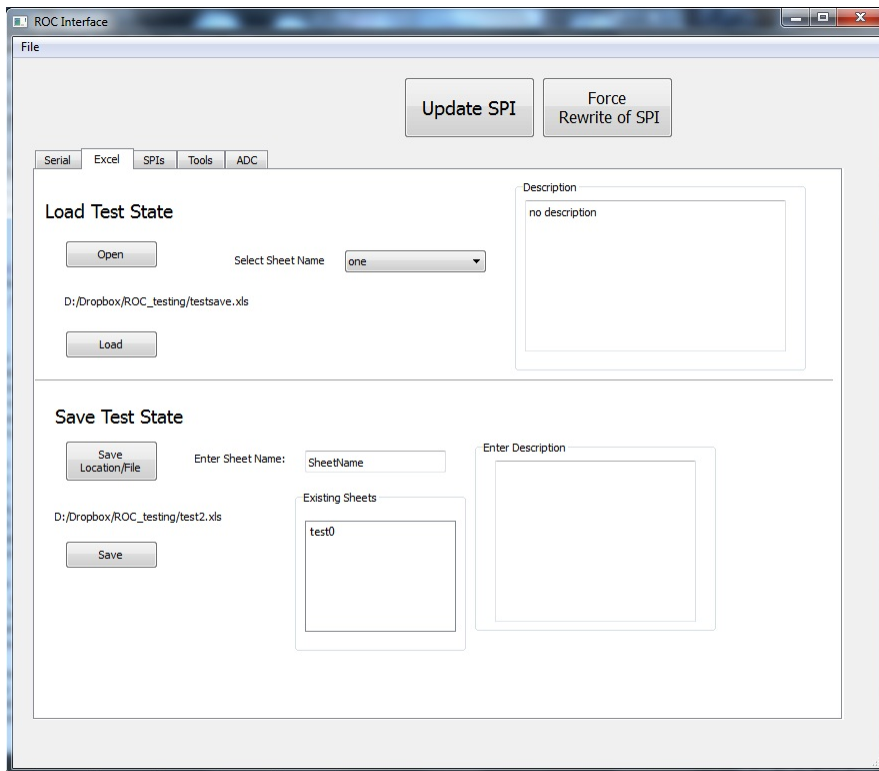


Figure 5.5: ROC Interface Excel tab

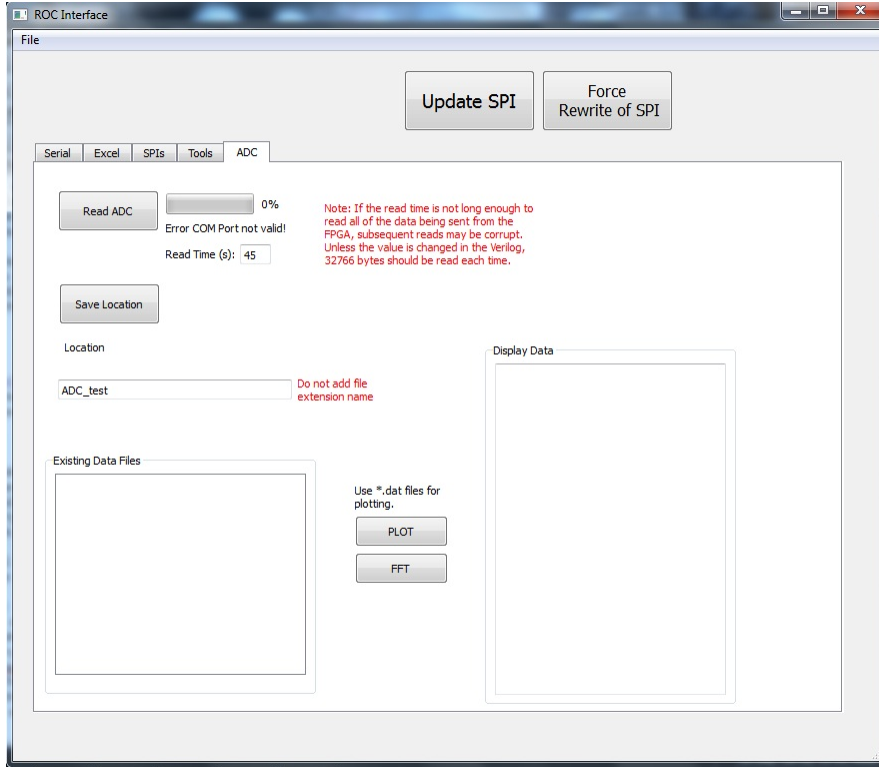


Figure 5.6: ROC Interface ADC tab

Future versions will calculate control bits to specify certain radar parameters such as PRF or range resolution.

5.4.6 ADC Tab

The ADC tab has all of the functions necessary for reading, saving, and viewing data from the ADC. When an ADC read command is sent to the ROC, the ADC data is first saved to memory on-board the SP601. The memory size can be set by editing the Spartan-6 Verilog. Once the data is saved on the SP601, it is transmitted to the host computer. The ADC read may take up to a minute, and the ADC tab has a status bar showing the progress of the read. The save location and file name may be selected for each ADC read. The current data or any saved data can be plotted in from the ADC tab. A screenshot of the ADC tab is given in Figure 5.6.

5.4.7 Packaging

Since Python and the third party packages are free, running the software on any computer can be accomplished by installing Python and the necessary packages. Although straightforward, installing Python and third party packages is time consuming. A list of the third party packages and their dependencies would also be required. Thankfully, there is a program named PyInstaller that converts Python programs to stand-alone executables. Once the program is thoroughly tested for crash scenarios, an executable is a very convenient way to package it.

Chapter 6

Summary

This chapter provides a summary of the work in the preceding chapters. It will highlight the key conclusions and accomplishments. It should provide a condensed reference for the rest of the chapters. A suggestion of future work concludes the chapter.

6.1 Simulation

The MATLAB simulation of the transceiver provides a quick analysis of the high-level details of the system. It can be easily modified to show changes in filters, frequencies, or stretch processing. Once the initial simulation is designed, the MATLAB simulation is faster and less error prone than hand calculations. The simulation is also much faster and easier to modify than a simulation in Cadence (the IC design software). The simulation can also model imperfections such as noise and interference.

6.2 Stretch Processing

Stretch processing is an alternative to matched filters. A matched filter correlates a received signal with the expected signal. The result is a pulse in time when the waveforms “line up.” A matched filter has optimum SNR. It is commonly implemented with SAW filters or in the digital domain. Every waveform has a unique matched filter. Multiple matched filters are required to vary the slope of an LFM waveform. A matched filter can be implemented digitally, but the bandwidth of a waveform is limited by the ADC.

Stretch processing compresses LFM waveforms into a pulse in frequency. It multiplies the incoming signal with a reference LFM waveform. Stretch processing can be actualized with mixers on the IC. This allows the ADC to have a much smaller bandwidth than the

transmitted waveform. The reference waveform can come from the same waveform generator that supplies the transmitted waveform.

Several constraints dictated how stretch processing could be designed for the ROC. One ADC was used on the ROC. This means that I-Q processing in the digital domain is not possible. The bandwidth of the ADC is also a limiting factor. The destretched or range frequency cannot exceed the bandwidth of the ADC. Since there is no digital I-Q processing, negative frequencies have to be distinguished from positive frequencies in the analog domain. Negative frequencies correspond to returns signals that arrive before the reference waveform starts. The ROC receiver resolves negative and positive ranges.

The stretch processing on the ROC has one serious problem. The destretching is performed by mixing the received baseband signal with the reference LFM waveform at baseband. The difference of the two waveforms is the single-tone signal. The sum of the signals is another LFM waveform at baseband. This sum or chirp-squared waveform usually does not affect the corresponding difference signal, but it can hide other signals that have lower power. The solution is to do the stretch processing at the IF. This requires quadrature IF+LFM waveforms. The in-phase waveform is already available. The quadrature phase signal is generated with a few extra mixer. The rest of the stretch processing remains the same. Figure 6.1 is the diagram for the improved implementation.

6.2.1 Analysis

This section summarizes the key conclusions and equations from the ROC stretch processing analyses. To start with, the basic results for stretch processing are

$$\Delta\tau = \frac{f_M}{\alpha} \tag{6.1}$$

$$\Delta r = \frac{c}{2}\Delta\tau \tag{6.2}$$

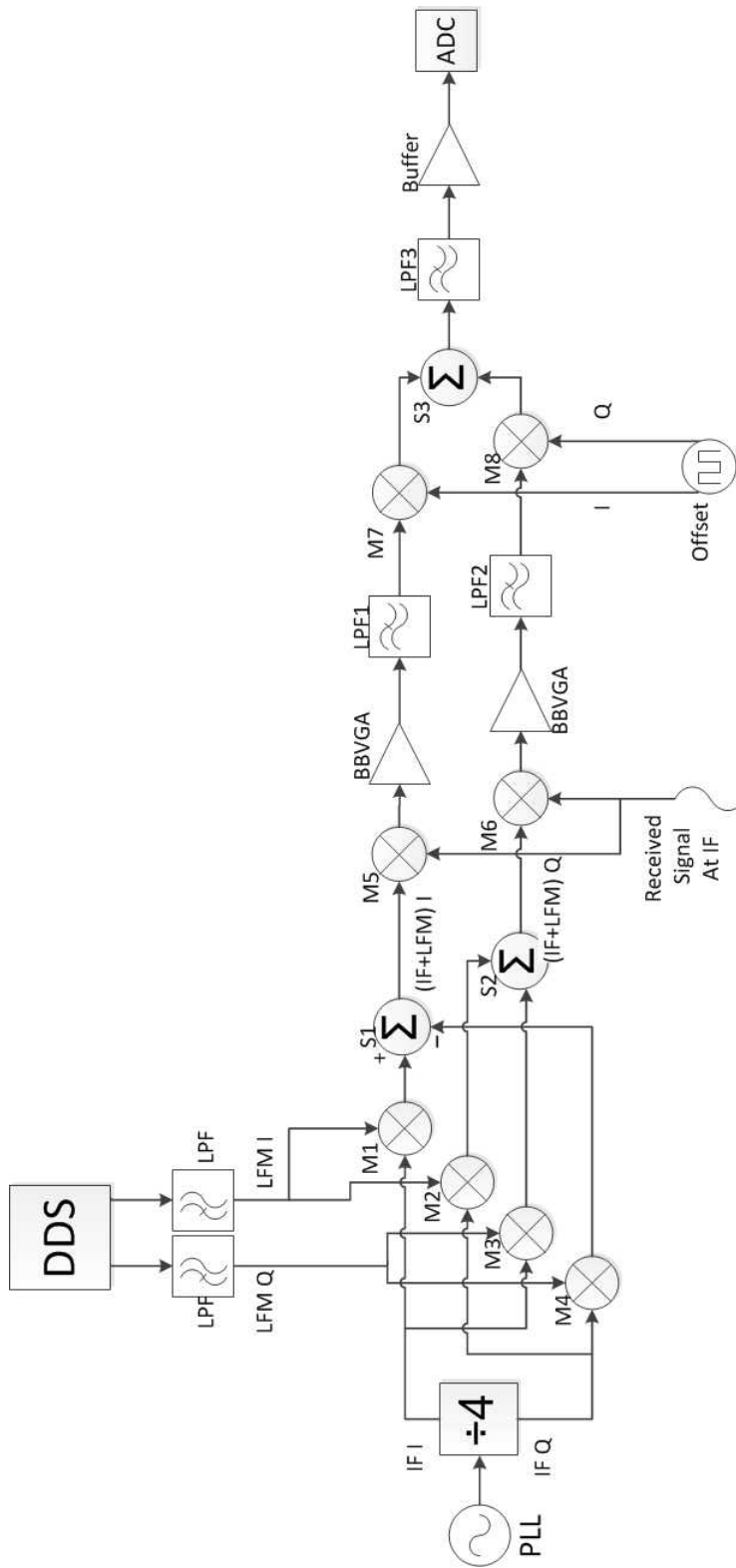


Figure 6.1: Updated Stretch Processing

where $\Delta\tau$ is the delay between the reference waveform and the incoming signal and Δr is the relative range and α is the slope of the LFM waveform.

The pulsewidth of the reference waveform and offset frequency need to be set to ensure that possible ranges are completely processed and returns from outside of the range window are not processed. The equations for the reference pulsewidth is

$$T = \text{time of transmitted pulse} + \frac{2f_{cutoff}}{\alpha} \quad (6.3)$$

The offset frequency should be equal to the cutoff frequency.

The range window is one of the biggest disadvantages of stretch processing. Stretch processing can only occur as long as the reference waveform is “on.” The reference waveform is limited either by the DDS or the system bandwidth capabilities. The equation for the range window is

$$\text{range window} = \frac{c}{2} \frac{f_{range}}{\alpha} \quad (6.4)$$

Doppler frequency directly adds to the range error of stretch processing. In most scenarios, the error is small though. The error cause by an object traveling with a radial velocity of v_r is

$$R_{dop} = \frac{v_r f_c}{\alpha} \quad (6.5)$$

where f_c is the carrier frequency.

Group delay variation is a measure of phase distortion. Wideband LFM waveforms are susceptible to group delay variation. Group delay causes a reduction in SNR and worse range resolution. The group delay spreads the target frequency. The power loss caused by group delay is

$$BW_{GD} = \alpha \Delta GD \quad (6.6)$$

$$P_{loss} = 10 \log \left(\frac{1}{T_{chirp} BW_{GD}} \right) \quad (6.7)$$

The range resolution because of group delay is

$$\Delta R = \frac{c}{2\alpha} \frac{1}{\text{BW}_{\text{GD}}} \quad (6.8)$$

I-Q imbalance is an issue for any quadrature processing. In the ROC stretch processing, the channel inequalities cause an image frequency to occur with a power of

$$P_{\text{image}} = \frac{A^2 - 2AB \cos(\theta) + B^2}{2} \quad (6.9)$$

where A and B are amplitudes and θ is the phase difference.

The processing gain is the ratio of output SNR to input SNR. Maximum processing gain is equal to the time-bandwidth product of the transmitted LFM waveform. It can be degraded if the noise bandwidth is larger than the LFM transmitted bandwidth. Where B_n is the noise bandwidth, B_r is reference LFM bandwidth and B_s is the transmitted LFM bandwidth:

- Same as matched filter for $B_s = B_n$
- Loss is proportional to B_r/B_s if the noise bandwidth is larger than the reference signal
- Loss proportional to B_n/B_s if the noise bandwidth is smaller than the reference bandwidth

6.3 Circuits

Group delay variation is worst at the band edges of filters and amplifiers. Group delay variation compensation can be had by the use of an equalizer. An LNA can be designed using narrowband LNA design techniques, and an equalizer can be added at the output to smooth gain and group delay without affecting noise figure.

A common base–common collector with voltage–voltage feedback is a possible alternative to a traditional cascode. It has tunable input matching, good high frequency performance,

and input matching that does not depend on emitter length. The circuit tested used shared DC current by using a tank to make the connection appear as an open to the frequency operating range.

6.4 Software

The ROC has hundreds of control bits that are controlled with a SPI. The ROC communicates with a Xilinx Spartan-6 development board. The development board appears as a COM port to the computer. Testing software needs to provide a quick intuitive way to communicate with the ROC. Python is the platform used for software development. It is free along with useful third party packages. PyQt allows for streamlined GUI development. The software has the following features:

- Communicates with the ROC
- Can update bits individually or as a register
- Save and load test states using Excel spreadsheets
- Read and view data from the ADC
- Built into an executable

6.5 Future Work

- If the ROC is re-fabricated the improved stretch processing should be used.
- The stretch processing analysis needs to be compiled into a more useful form by making a software tool based on it. The user can input some system features such as range resolution and maximum bandwidth, and the other parameters are calculated accordingly.

- The testing software can be updated to be able to control radar operation. Instead of a user entering control bits, radar parameters like PRF and range resolution are defined.
- The integrated duplexer is an interesting circuit. Further research and maybe fabrication needs to be done to see if it really is viable.

Bibliography

- [1] Skolnik, Merrill. *Radar Handbook*. 3rd ed. McGraw-Hill, 2008.
- [2] T. Long, Y. Wang and T. Zeng, "Signal-to-noise ratio in stretch processing," *Electronics Letters* 13th May 2010 Vol. 46 No. 10
- [3] M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar: Basic Principles*. Raleigh, NC: SciTech Publishing, 2010.
- [4] Caputi W.J., "Stretch: a time-transformation technique," *IEEE Trans Aerosp, Electronics, Systems*, 1971, 7, pp. 269-278
- [5] Kuo, W.-M.L.; Qingqing Liang; Cressler, J.D.; Mitchell, M.A.; , "An X-band SiGe LNA with 1.36 dB mean noise figure for monolithic phased array transmit/receive radar modules," *Radio Frequency Integrated Circuits (RFIC) Symposium*, 2006 IEEE , vol., no., pp.4 pp.-501, 11-13 June 2006
- [6] Rossi, P.; Liscidini, A.; Brandolini, M.; Svelto, F.; , "A variable gain RF front-end, based on a Voltage-Voltage feedback LNA, for multistandard applications," *Solid-State Circuits, IEEE Journal of* , vol.40, no.3, pp. 690- 697, March 2005
- [7] Gray, P.R., et al. *Analysis and Design of Analog Integrated Circuits*. 4th ed. John Wiley and Sons, Inc, 2001
- [8] Mikhemar, M.; Darabi, H.; Abidi, A.; , "An on-chip wideband and low-loss duplexer for 3G/4G CMOS radios," *VLSI Circuits (VLSIC), 2010 IEEE Symposium on* , vol., no., pp.129-130, 16-18 June 2010
- [9] Darabi, H.; , "Highly integrated and tunable RF front-ends for reconfigurable multi-band transceivers," *Custom Integrated Circuits Conference (CICC), 2010 IEEE* , vol., no., pp.1-8, 19-22 Sept. 2010
- [10] Sartori, E.; , "Hybrid Transformers," *Parts, Materials and Packaging, IEEE Transactions on* , vol.4, no.3, pp. 59- 66, Sep 1968
- [11] Thornber, Martin. "Amplitude Equalizer Flattens Gain from 50 MHz to 7 GHz." *MicroWaves and RF*. October 2007 <www.mwrf.com/Articles/ArticleID/17224/17224.html>
- [12] "Python v2.7.2 Documentation." Python Software Foundation. February 2012 <docs.python.org>

- [13] Wentworth, S.M. *Applied Electromagnetics: Early Transmission Lines Approach*. Wiley, 2007. pp. 592-602