WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ELECTRONICS

FIELD: Elektronika (EKA)
SPECIALIZATION: Advanced Applied Electronics(AAE)

MASTER OF SCIENCE THESIS

Airplane tracking system using ADS-B

System lokalizacji samolotów z wykorzystaniem ADS-B

AUTHOR:
Karol Szpila

SUPERVISOR:

Ph.D., D.Sc. Grzegorz Budzyń

Institute of Telecommunications, Teleinformatics and Acoustics (I-28)

GRADE:

WROCŁAW 2019

Nomenclature

ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
FPGA	Field Programmable Gate Array
SDR	Software Defined Radio
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
SoC	System on Chip
SPI	*UART*

Contents

Chapter 1

Introduction

1.1 Purpose and aim

The purpose of this paper is to study various parameters defining RF signal quality and models of IQ imbalance. Research concept of Software Defined Radio, principles of operation of such devices and capability of Xilinx Zynq SoC in such domain. Evaluate different correction algorithms implementation. Compare it performance for various type of signals: singletone, multitone, broadband, and 4-QAM, 8-QAM8 and 16-QAM modulated. The comparison include simulation in Matlab, implementation hardware (FPGA part of ZYNQ SoC) and native correction inside RF transceiver chip.

1.2 Thesis outline

Chapter 2

Theoretical background

In this chapter the theoretical operation of basic RF fronted components is explained. This elements are mixers, quadrature modulators and demodulators. Next, widely used in RF communication, IQ signal model is described, together with two imbalance models. After that Software Defined Radio concept is presented.

2.1 Theoretical operation of RF mixer

The mixer is one of the basics component used in RF communication. It is nonlinear circuit used for signal multiplication. Such process is used virtually in every radio system, cellular base stations, radars and many more devices.

RF mixer is three port devices, which takes as an input two signal and produces signal consisting of the two frequencies on the output. Due to the non linearity of the mixer the new signal on the different frequency is generated.

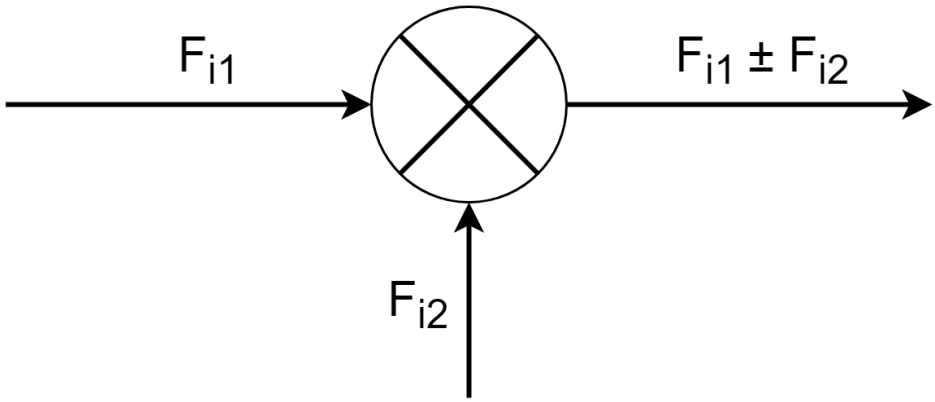

Figure 2.1: *Schematic of RF mixer.*

Where:

- F_{i1} and F_{i2} are input signal that are mixed together.
- $F_i1 + F_i2$ is USB (Upper Side Band) output signal,
- $F_i1 - F_i2$ is LSB (Lower Side Band) output signal.

In every mixing operation USB and LSB signal are generated. Depending on the relation between RF and LO signal one of the band is desired output and other is undesired image.

When $RF < LS$, mixer works as up-converter USB is desired and LSB is undesired.

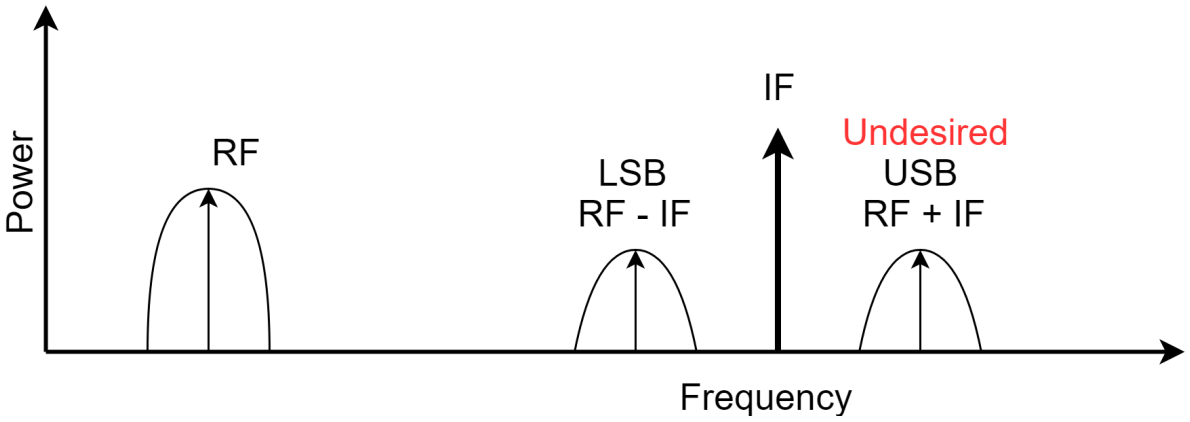

Figure 2.2: *Schematic of RF mixer.*

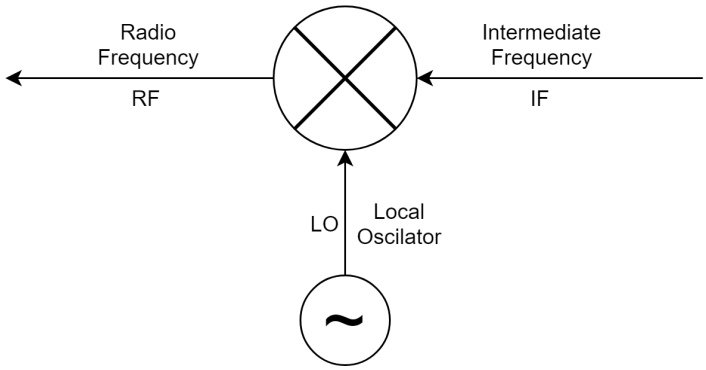

Figure 2.3: *Schematic of RF mixer working as up-converter.*

When $RF < LS$, mixer works as down-converter LSB is desired and USB is undesired..

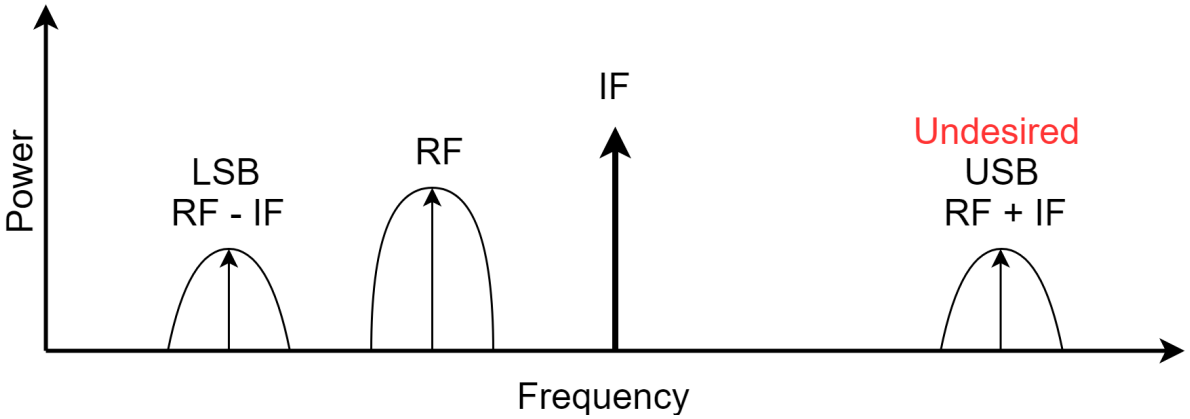

Figure 2.4: *Schematic of RF mixer.*

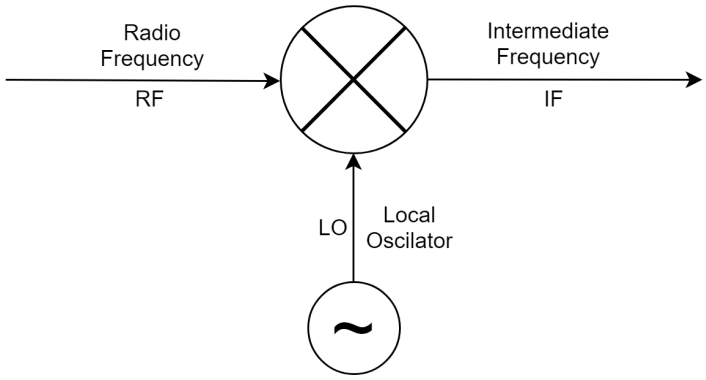

Figure 2.5: *Schematic of RF mixer working as down-converter.*

To remove necessary to include in the RF mixer design BPFs (Band Pass Filters). This filter attenuates all frequencies outside specified band.

Application for mixers are mostly:

- Frequency shifting - The most common application for RF mixer is changing signal frequency with preservation of others parameters such as amplitude and phase. This technology is mostly used in RF transmitter and receivers. Such devices are called up-converters (for mixing signal with carrier) and down-converter for (extracting signal from carrier).
- Phase comparison - It is possible to detect phase difference between two signal using mixer. This RF mixer application is used in PLLs (Phase Locked Loops).

2.2 IQ signal model

The term IQ is an abbreviation for in-phase and quadrature. Signal are considered in-phase when phase of both is equal and quadrature when it differs by 90deg. IQ data model shows changes in phase and magnitude of a sine wave. Modification of these parameters allow to encode information upon a sine wave.

Equation of the sine wave is:

$$A cos\left(2\pi f t + \phi\right),$$

where:

- A is amplitude,
- f is frequency,
- ϕ is phase shift

According to equation only amplitude, phase and frequency of the sine wave can be modified. Moreover frequency is first derivative of phase. Therefore it can be collectively referred to as the phase angle. According to these assumptions the instantaneous state of a sine wave can be described in complex plain using magnitude and phase as polar coordinates.

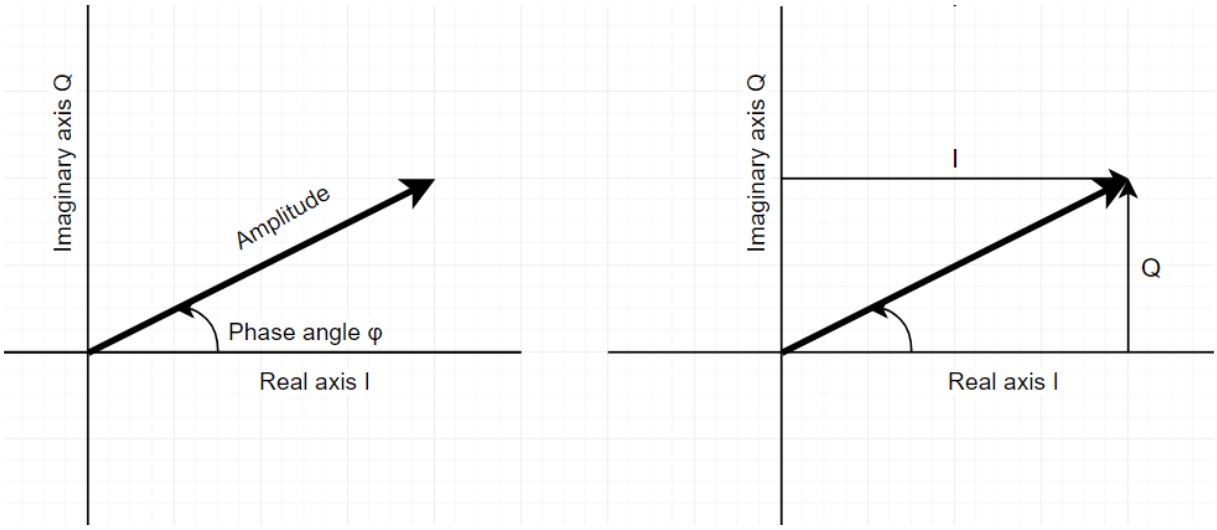

Figure 2.6: *Representation of sine wave in complex plain*

Using trigonometry, the polar coordinates can be converted into I and Q components of the signal using equations:

- $I = Acos\,(2\pi ft)$,
- $Q = Asin\,(2\pi ft)$,

IQ data model is widely used in RF communication systems. It allows to distinguish type of modulation used on carrier. Allows to introduce concept of positive and negative frequency. Amplitude and phase angle form seems to be more intuitive, however precisely varying the phase of a high-frequency carrier sine wave in a hardware circuit according to an input message signal is difficult. Therefore such hardware modulators will be expensive and hard to design and build. To avoid direct modulation of RF signal phase signal is decomposed to I and Q components.

According to Ptolemy's identitie for the cosine of sum:

$$cos\,(x+y) = cos\,(x)\,cos\,(y) - sin\,(x)\,sin\,(y)$$

sine wave carrier can be represented as:

$$Acos\,(2\pi ft + \phi) = Acos\,(2\pi ft)\,cos\,(\phi) - Asin\,(2\pi)\,.ft)sin(\phi)$$

Using equation 2.2 following formula is obtained:

$$Acos\,(2\pi ft + \phi) = Icos\,(2\pi ft) - Qsin\,(2\pi ft)\,,$$

where:

- I - is amplitude of in-phase signal,
- Q - is amplitude of quadrature signal.

Using this data samples representation, modulation of phase of the RF signal is possible just by modulation of I/Q signals amplitudes and then mix it with carrier and quadrature of carrier using mixers. Schematics below shows structure of IQ modulator and demodulator.

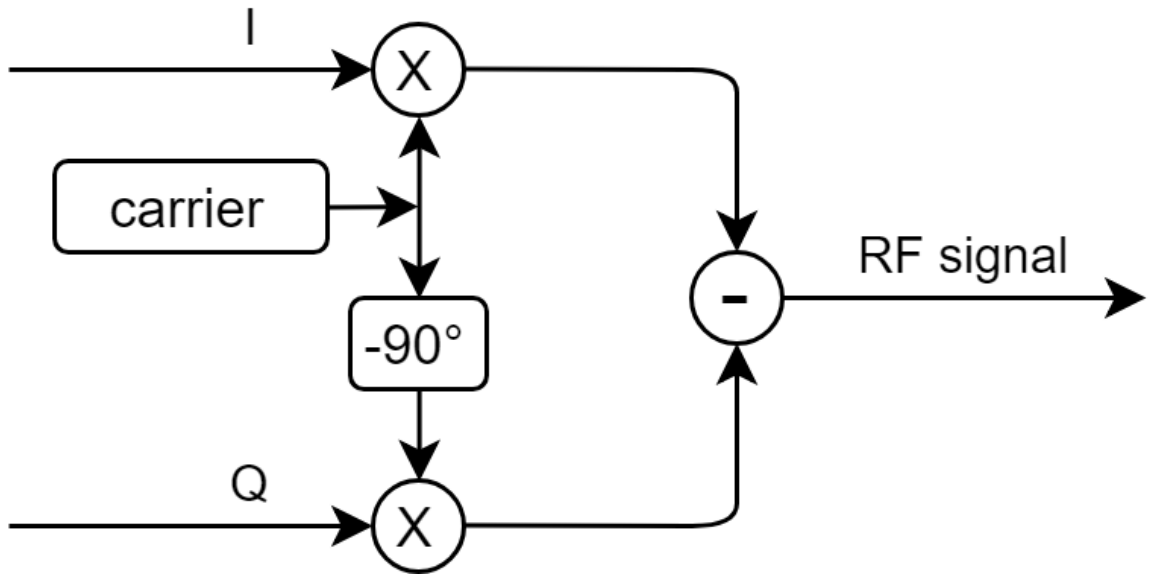

Figure 2.7: *Schematic of IQ modulator*

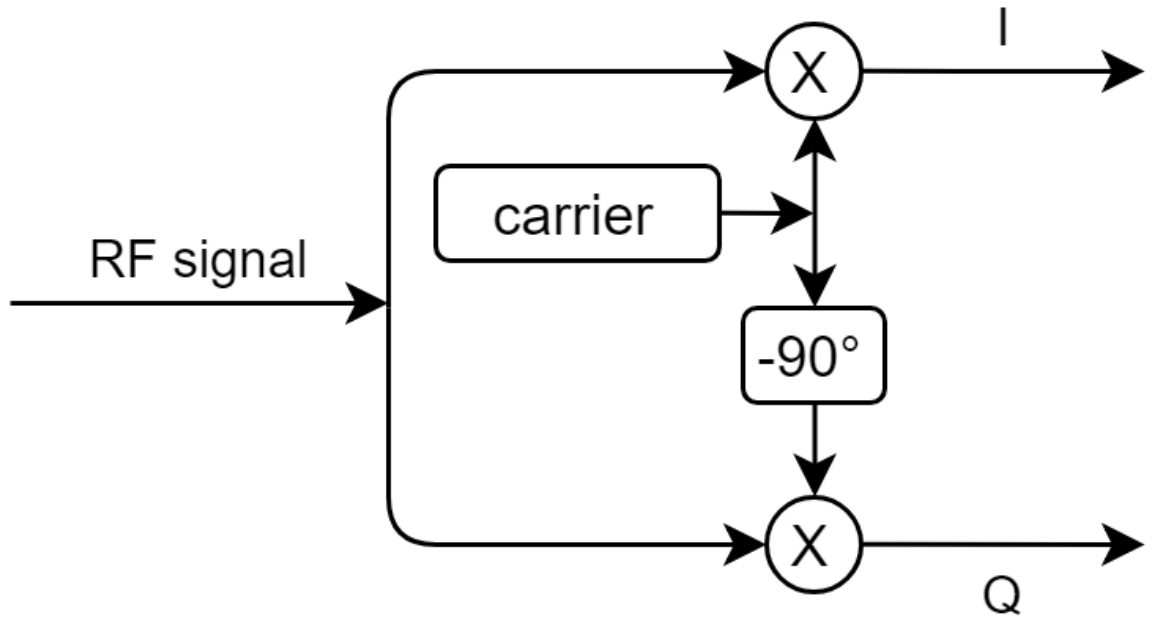

Figure 2.8: *Schematic of IQ demodulator*

The flexibility and simplicity of this solution compared to direct phase manipulations is a reason why I/Q modulators and demodulators are so widely used and popular in RF hardware.

2.3 IQ imbalance models

In this chapter, two IQ imbalance models are described. First is called SBIQM (Single-Branch IQ Imbalance Model). In this model imbalance exist only in one branch. Second is called DBIQM (Dingle-Branch IQ Imbalance Model). Algorithms presented in this thesis are based on DBIQM model. Mentioned model assumes imbalance in both I and Q branch.

2.4 Software Defined Radio

SDR (Software Defined Radio) is a radio communication system where components typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators), are instead implemented by the means of software.

Below the block diagram of SDR device is presented:

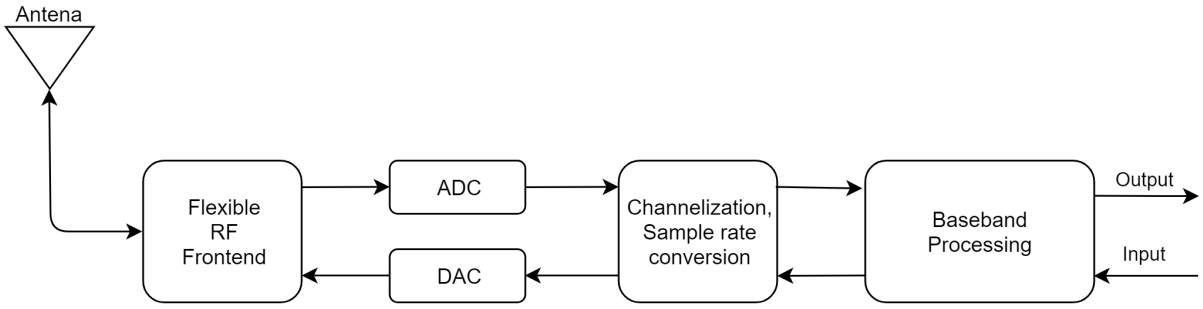

Figure 2.9: *Block diagram of SDR device.*

- **Flexible RF fronted** is a first stage of SDR processing system. This part is configured by the means software to match current requirements for best system performance. This is possible by using components numerically controlled components. NCO (Numerically Controled Oscilators) which are used as flexible frequency generators. This component are essential for I/Q modulators/demodulators as flexible carrier generators for mixers. Bandpass filters and equalizers used for signal conditioning.

- **ADC / DAC converters**. This is where captured signal is converted from analog to digital domain and likewise transmited is converted form digital to analog domain. This is a last stage before before purely digital domain of the system.

- **Channelization and Sample rate conversion**. Here the data processed by the baseband system are assigned to respective communication channels, and conditioned for specific channel transmission parameters like frequency and sampling rate.

- **Baseband Processing System** Digital core of the SDR system. This may by either processor, FPGA, ASIC or SoC. Variety of choice in this case allows creation of well-suited system for desired performance and cost range. Mentioned component is responsible with interfacing with external system for data exchange purpose. All logic related to transmission is implemented here in the software including correction algorithms and application layer. Thanks to that system can adapt to changes. This means improving signal processing software to improve quality of the processed signal or even complete change of the application layer, by simply providing new software.

Chapter 3

Hardware and tools

This chapter describes chosen hardware, tools and explains reasons behind such decisions.

3.1 Adalm Pluto and AD tools

Adalm Pluto Board

Adalm Pluto is learning module from Analog Devices that can be used to introduce principles of operation and theory behind SDR and RF communication.

Figure 3.1: *Image of Adalm Pluto device.*

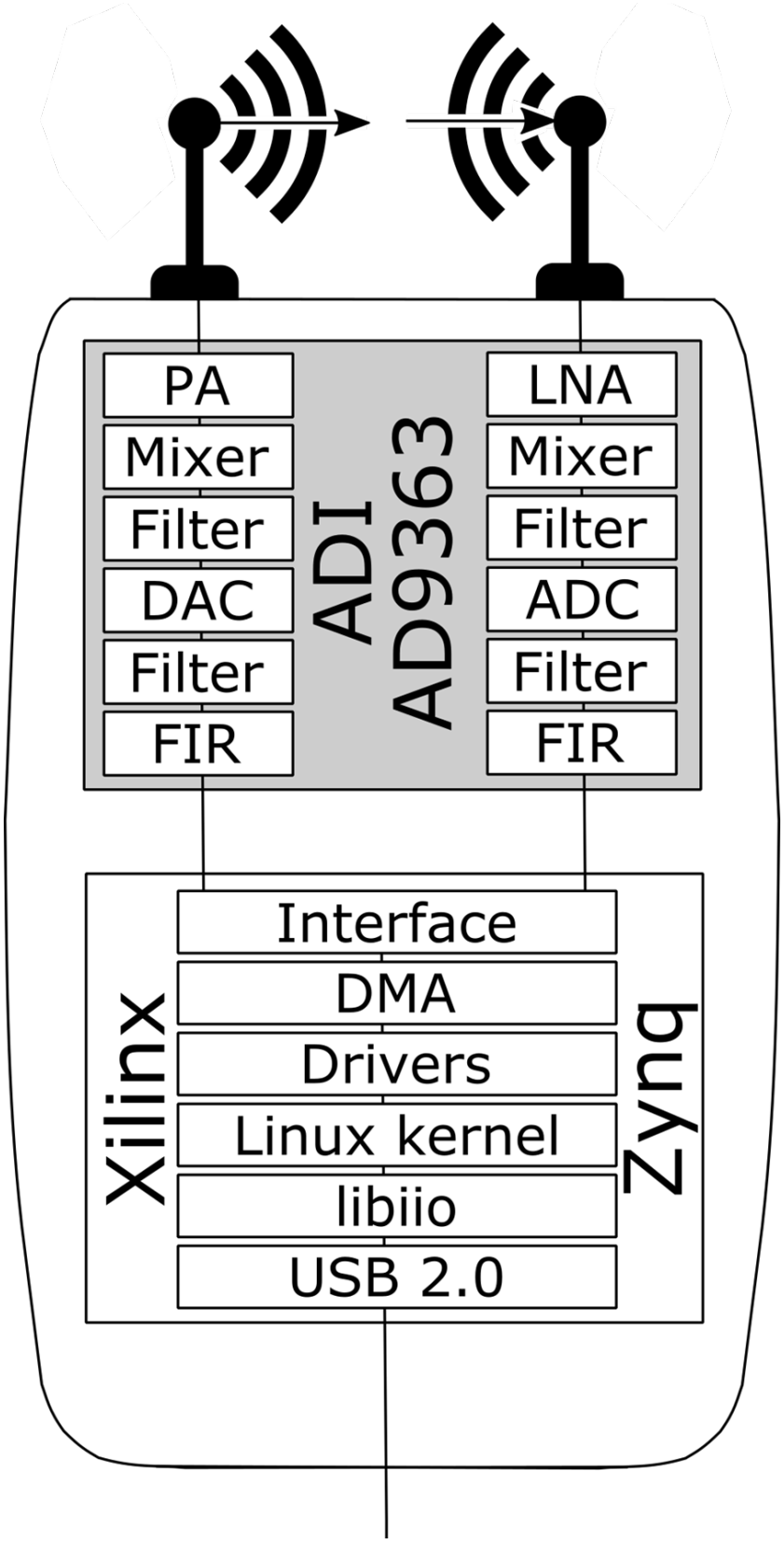

Figure 3.2: *Schematic Adalm Pluto hardware*

Adalm Pluto is already integrated with with MATLAB, Simulink, GNU Radio, and libIIO allows to interface with the devices using C, C++, C# or Python programming languages. PlutoSDR is open software repository with all software and tools related to Adalm Pluto including: HDL Project in Xillinx Vivado, Buildroot configuration for

embedded Linux. Main reason for choosing this SDR device was relatively low price in comparative to other available solutions on the market. All other devices were FPGA based. Adalm Pluto core is Zynq7010 SoC which is combination of ARM based processor capable of running Linux and FPGA part which can communicate with the processor via AXI interface.

AD9363 Agile RF Transceiver

The main part of Adalm Pluto SDR is AD9363. AD9363 is a high performance, highly integrated RF agile transceiver designed for use in 3G and 4G femtocell applications.

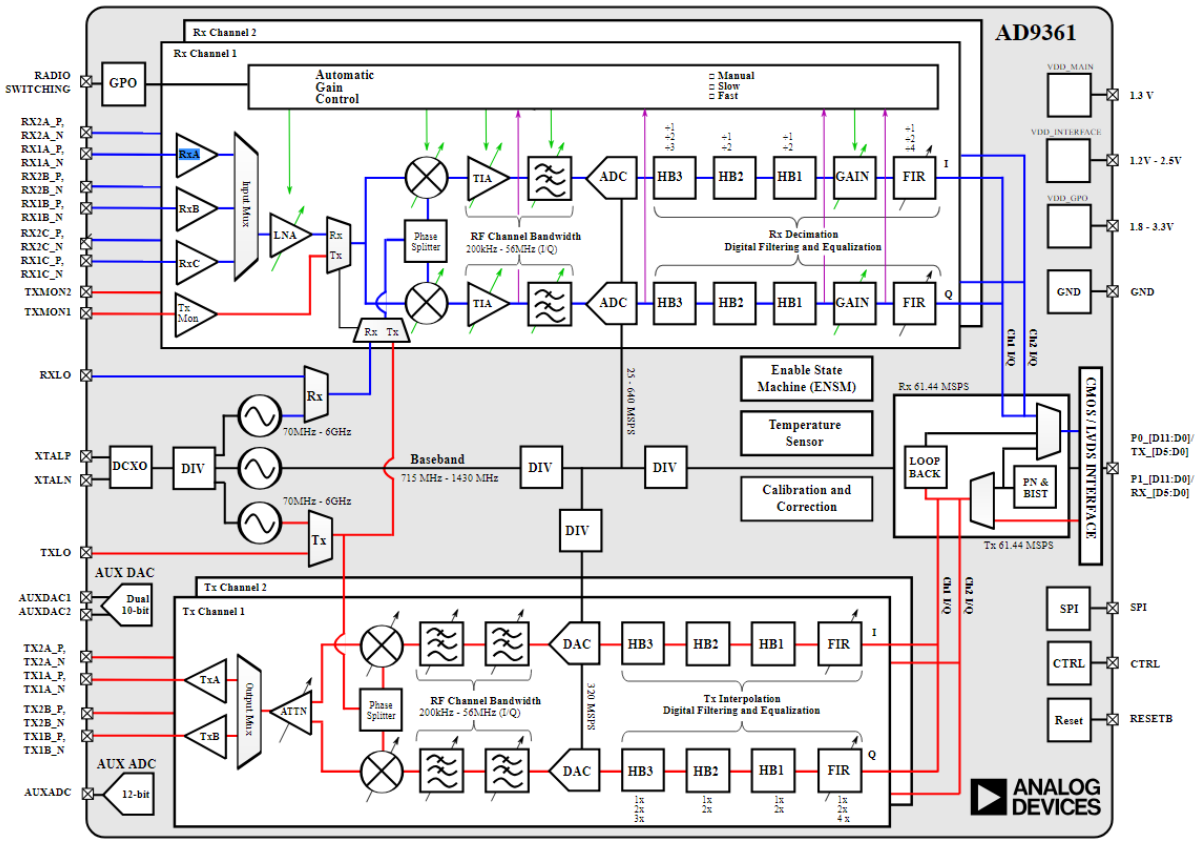

Figure 3.3: *Block diagram of AD9363 RF transceiver*

Ad9363 features:

- Wide bandwidth from 325Mhz to 3.8GHz,
- Tunnable channel bandwidth up to 20MHz,
- Independent AGC (Automatic Gain Control),
- Full duplex communication,
- DC offset and I/Q mismatch tracking.
- Flexible rate, 12-bit ADC and DAC

IIO Osciloscope

The ADI IIO Oscilloscope is application which allows interfacing with different evaluation or custom made board based on AD agile RF Transrecivers. This program allows communication with the device connected with PC using:

- Local or remote network,
- USB 2.0 interface,
- Serial Port.

Application supports full configuration of the device including: configuration receiver frequency and channel bandwidth. Access to DC offset tracking, I/Q correction, internal calibration functionality and all registers. Moreover allows to process revived using filters designed in matlab and four gain control modes:

- manual configuration - gain is selected by user.
- slow attack - gain is selected to attenuate backgrund noise as much as possible,
- fast attack
- hybrid attack.

Program allows to plot received data from selected channels in four different modes:

- time domain as I and Q signals,
- frequency domain with configurable FFT and averaging size,
- constellation as relation between I and Q signals,
- cross-correlation for multi-channel boards.

3.2 libIIO

libIIO is library developed by Analog devices for interfacing Linux IIO (Industrial Input Output) devices.

Library is composed of three main parts:

- local backend - responsible for interfacing with the Linux kernel using sysfs virtual filesystem,
- network backend - which comunicatios tiwh the IIOD (IIO Deamon) by network link.

The IIOD is libiio based application that creates libiio contex that is using local backend and then connects it to network backend. Described process allows to hare local context with the server and allows interfacing with the evice using network.

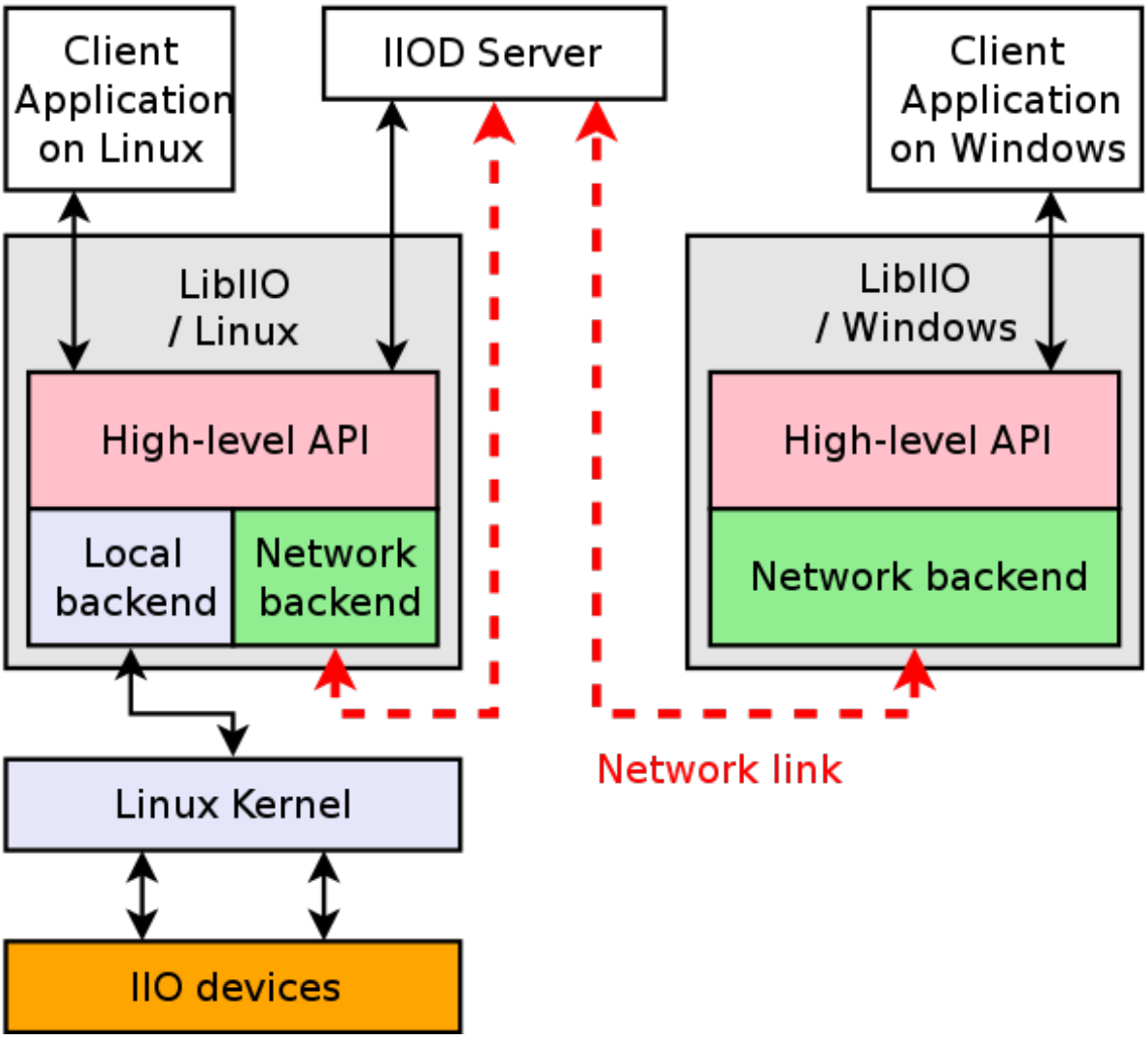

Figure 3.4: *Block diagram of AD9363 RF transceiver*

3.3 Zynq and Xilinx tools

Introduction to SoC

SoC is aberration from System on Chip. Such devices aggregates many types of hardware in single chips including: ADCs, DACs, internal memory, external memory controllers, peripheral interfaces such as SPI, UART, I2C interfaces controllers and many others. The main idea is that single silicon chip may be used to implement functionality of entire system. This is cheaper and faster solution than realizing such functionality on PCB using separate components. In the past such role belonged to ASIC. (Application Specific Integrated Circuits). The major disadvantages of such solution is lack of flexibility and significant development cost and time. This makes ASIC's sustainable only on the high volume market where no future upgrade will be required. These limitations created clear need for more flexible device with faster development time. This need has been long satisfied by a FPGAs. FPGA can be reconfigured as desired which means virtually no risk in deploying solution which may require upgrade based on FPGA. Next step are SoC based solution. SoC is combination of processor and FPGA. This allows to create

fast application dependent hardware functionality. Moreover processor can run normal operating system which allows fast development and flexibility of the solution.

Zynq-7000 family SoC

The Zynq is new kind of SoC from Xillinx. It combines both applications processor and FPGA fabric. The Zynq device comprises two sections: PS (Processing System) and PL (Programmable Logic). This sections can be used separately for independent task or together to utilize advantages of both software and hardware. The Zynq devices are meant to use structure of both sections and interface between them. Connection between these parts is provided by AXI (Advanced eXtensible Interface) which is registered under ARM trademark.

Processing System in all Zynq devices has the same architecture, and the base of it is a dual-core ARM Cortex-A9 processor. This is a hard processor which means it is manufactured directly in silicon structure. Zynq allows to use soft processors like PicoBlaze and MicroBlaze which can be implemented in PL sections. Depending on the size and speedgrade of the device ARM processor is accompanied by set of processing resources e.g (MMU, DMA, SRAM, Processing System External Interfaces, cache memory, control registers , etc.) which forms together APU (Application Processing Unit).

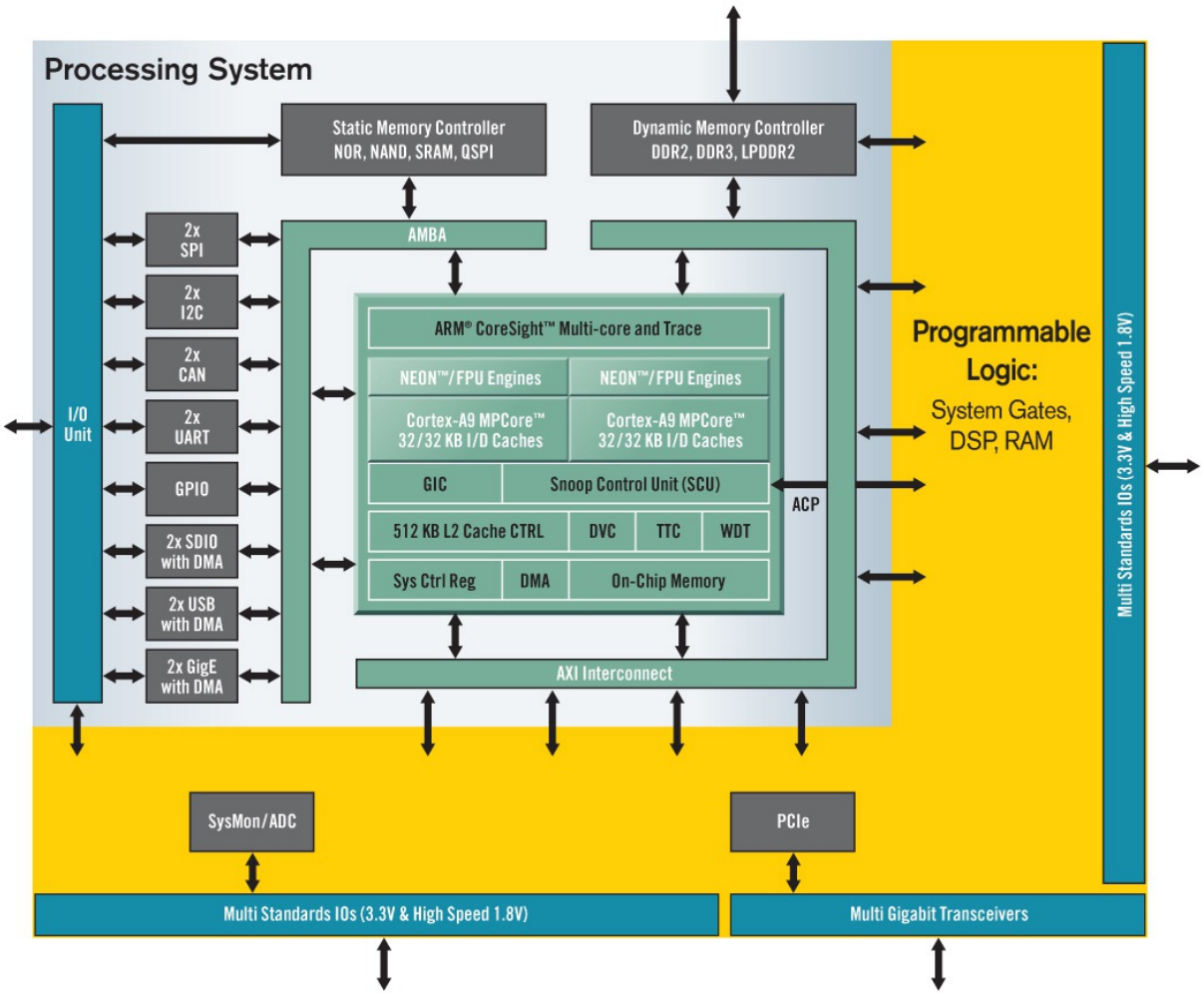

Figure 3.5: *Block diagram fo Zynq 7000 family SoC*

Programmable Logic is a FPGA based on Atrix-7 and Kintex-7 fabric from Xillinx. PL consist of CLBs (Configurable Logic Blocks), slices, IOBs(Input Output Blocks), LUTs (Lookup tables), fliplops and switch matrices. Additionally there are BRAM (Block Random Access Memory) which are used to store large amout of date in small part of the device which allows fast access to its content. Second additional resource is DSP48E1 slice. This is advanced DSP module that allows many complex mathematical operation in a single clock cycle.

Xillinx Vivado Tools

As a part of Zynq-7000 family SoC design flow. Xillinx Vivado 2018.2 and Xilinx Vivado HLS 2018.2 where choosen. This version of the software is tested and compatible with latest release of the plutosdr-fw repository.

Xillinx Vivado is a tool for FPGA design. This program allows full configuration of PL part of the Zynq-7000 family SoC including: synthesis, implementation, routing, timing validation and generation of the bitstream. Bitstream is used to deploy configuration onto FPGA device.

Xilinx Vivado HLS is a program for High Level Synthesis for Zynq-7000 family SoC. This tool allows to write and test software in C language. After proper validation in the software domain, code can be synthesized to HDL and exported to Xillinx Vivado as an IP Core. This allows much faster design than direct HDL development and much easier testing. These were main reason for choosing HLS environment.

Chapter 4

Algorithms

This chapter describes all algorithms used for signal quality improvement. All of considered algorithms are blind. This means that signal analysis is purely statistical with no prior information about signal. Advantages of this approach is that this algorithms can be used for any type of possible signal.

4.1 DC offset correction

This section describes algorithms used for removing DC offset from revived samples.

Moving Average Filter

Moving Average Filter is simple FIR (Finite Impulse Response) filter. This filter is commonly used for white noise removal and signal smoothing. However when filter order is great enough to account for samples from full period of the signal filters the average represents signal DC offset.

The filter equations is presented below:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k], \tag{4.1}$$

where:

- N is order of the filter.
- $y[n]$ is filtered sample at n-th step,
- $x[n]$ is n-th sample from receiver.

Corrected values are calculated as follows:

$$I/Q_{corr} = I/Q - I/Q_{mean},$$

where:

- I/Q is recived I/Q sample,
- I/Q_{mean} is DC offset calculated using Moving Average Filter.

Algorithm introduces delay by N samples in the signal. It is important to that the number of samples taken into consideration must extend at least one period of the signal. Otherwise filter will works white noise attenuator.

Normalized Gaussian Filter

Normalized Gaussian Filter is filter whose impulse response has shape of Gaussian function with all values in range from 0 to 1.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x}{2\sigma^2}}$$

where:

- x is
- σ is standard deviation.

The filter equations is presented below:

$$y[n] = \sum_{k=0}^{N-1} x[n-k]G(x) \tag{4.2}$$

- N is windows size,

Corrected values are calculated as follows:

$$I/Q_{corr} = I/Q - I/Q_{gauss},$$

where:

- I/Q is recived I/Q sample,
- I/Q_{gauss} is DC offset calculated using Normalized Gauss Filter.

This filter has better performance in frequency domain than Moving Average Filter.

4.2 Magnitude and Phase Correction

The I/Q imbalance is common problem in RF front-ends that uses analog quadrature down- mixing. The ideal down-converter performs only simple frequency shift. Real down- converters introduces image interference which may be assumed as constant. But amplifiers and filters introduces varying with frequency imbalance.

In ideal case the receiver output is:

$$\begin{aligned} I(t) &= cos(\omega t) \\ Q(t) &= sin(\omega t) \end{aligned} \tag{4.3}$$

where:

- ω is tone frequency.

I and Q components are orthogonal with respect to each other.

In real case, assuming Single Branch Model receiver output is:

$$\begin{aligned} I(t)' &= \alpha cos(\omega t) + \beta_I \\ Q(t)' &= sin(\omega t + \phi) + \beta_Q \end{aligned} \tag{4.4}$$

where:

- α is magnitude mismatch,
- ϕ is phase imbalance,
- $\beta_{I/Q}$ is signal DC offset.

Input of phase correction algorithms is signal with DC offset extracted using algortihms from previews section. Hence the signal model is:

$$\begin{aligned} I(t)'' &= \alpha cos(\omega t) \\ Q(t)'' &= sin(\omega t + \phi) \end{aligned} \tag{4.5}$$

According to Ptolemy's identitie for the sine of sum:

$$sin\left(\omega t + \phi\right) = sin\left(\omega t\right))cos\left(\phi\right) + cos\left(\omega t\right) sin\left(\phi\right)$$

Equation **??** be rewritten in matrix form as:

$$\begin{bmatrix} I(t)'' \\ Q(t)'' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} \tag{4.6}$$

After multiplying both sides of **??** by inversion of parameters matrix following set of exuations is obtained:

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} \alpha^{-1} & 0 \\ \alpha^{-1} tan(\phi) & sec(\phi) \end{bmatrix} \begin{bmatrix} I''(t) \\ Q''(t) \end{bmatrix} \tag{4.7}$$

This shows that only α and ϕ must be found to perform I/Q mismatch compensation. According to this paper:

$$< I''(t)I''(t) >= \frac{1}{2}\alpha^2$$

$$\alpha = \sqrt{2 < I''(t)I''(t) >}$$

$$< I''(t)Q''(t) >= \frac{1}{2}\alpha^2 sin(\phi)$$

$$sin(\phi) = \frac{2}{\alpha} < I''(t)Q''(t) >$$

Assuming that $|\phi| \leq \frac{pi}{4}$ $cos(\phi)$ can be obtained directly from $\sin(\phi)$ using following formula:

$$cos(\phi) = \sqrt{1 - sin(\phi)}$$

Using following parameters in we can substitute matrix equations 4.6 with:

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & 0 \\ \frac{-sin(\phi)}{acos(\phi)} & \frac{1}{cos(\phi)} \end{bmatrix} \begin{bmatrix} I''(t) \\ Q''(t) \end{bmatrix} \tag{4.8}$$

The I/Q mismatch correction can be now applied using 4.8 formula.

Chapter 5

Simulations

This chapter contains description of test environment, matlab algorithms implementation and its performance evaluation

Moving Average Filter

This algorithm was directly implemented using 4.1, with order of the filter as one of the input parameters.

```
function [ x_corr ] = moving_average_fir( x, order )

    x_corr = zeros(1,length(x)    order)';
    for i=1 : length(x_corr)
        prev_x = x(i:i+order    1);
        x_corr(i) = sum(prev_x)/order;
    end
end
```

Normalized Gaussian Filter

Presented filter is direct implementation of 4.2 equation, with window siz and standard deviations as input parameters.

```
function [ x_corr ] = gauss_smooth( x, sigma, window_size )

    sigma = 5;
    window_size = 30;    % length of gaussFilter vector
    x_corr = zeros(1,length(x)    window_size);
    y = linspace( window_size / 2, window_size / 2, window_size);
    gaussFilter = exp( y .^ 2 / (2 * sigma ^ 2));
    gaussFilter = gaussFilter / sum (gaussFilter); % normalize

    for i=1 : length(x_corr)    window_size
        prev_x = x(i:i+window_size 1);
        x_corr(i) = sum(gaussFilter.*prev_x');
    end
    x_corr = x_corr';
end
```

Magnitude and Phase Correction

Presented filter is direct implementation of 4.2 equation, with window siz and standard deviations as input parameters.

```
function [ Iend, Qend ] = iq_corr( I, Q )
    step = 256;
    range = 4096;
    bI = moving_average_fir( I, step);
    bQ = moving_average_fir( Q, step);

    %bI = gauss_smooth( I, sigma, step);
    %bQ = gauss_smooth( Q, sigma, step);

    I_corr = I(step+1:end)    bI;
    Q_corr = Q(step+1:end)    bQ;


    Ipp = moving_average_fir( I_corr.*I_corr, step )/(range.^2);
    %Ipp = gauss_smooth( I_corr.*I_corr, sigma, step);
    alpha = sqrt(2*Ipp);

    IQpp = moving_average_fir(I_corr.*Q_corr, step)/(range.^2);
    %IQpp = gauss_smooth( I_corr.*Q_corr, sigma, step);
    sinw = (2./alpha) .* IQpp;
    cosw = sqrt(1    sinw.*sinw);

    A = 1./alpha;
    C = ( sinw)./(alpha.*cosw);
    D = 1./cosw;

    Iend = A.*I_corr(step+1:end);
    Qend = A.*(C.*I_corr(step+1:end) + D.*Q_corr(step+1:end));
end
```

Figure 5.1 presents flow of the test application in Matlab environment. IQ samples are first processed by DC offset removal algorithm depending one the value of dc_corr_type variable, by either Moving Average Filter or Normalized Gaussian Filter. Signal after this correction are available in the workspace under *Ip* and *Qp* names. Next signal are processed by Magnitude and Phase Correction algorithm. *Ipp* and *Qpp* names of variables holding signal samples after full I/Q mismatch compensation.

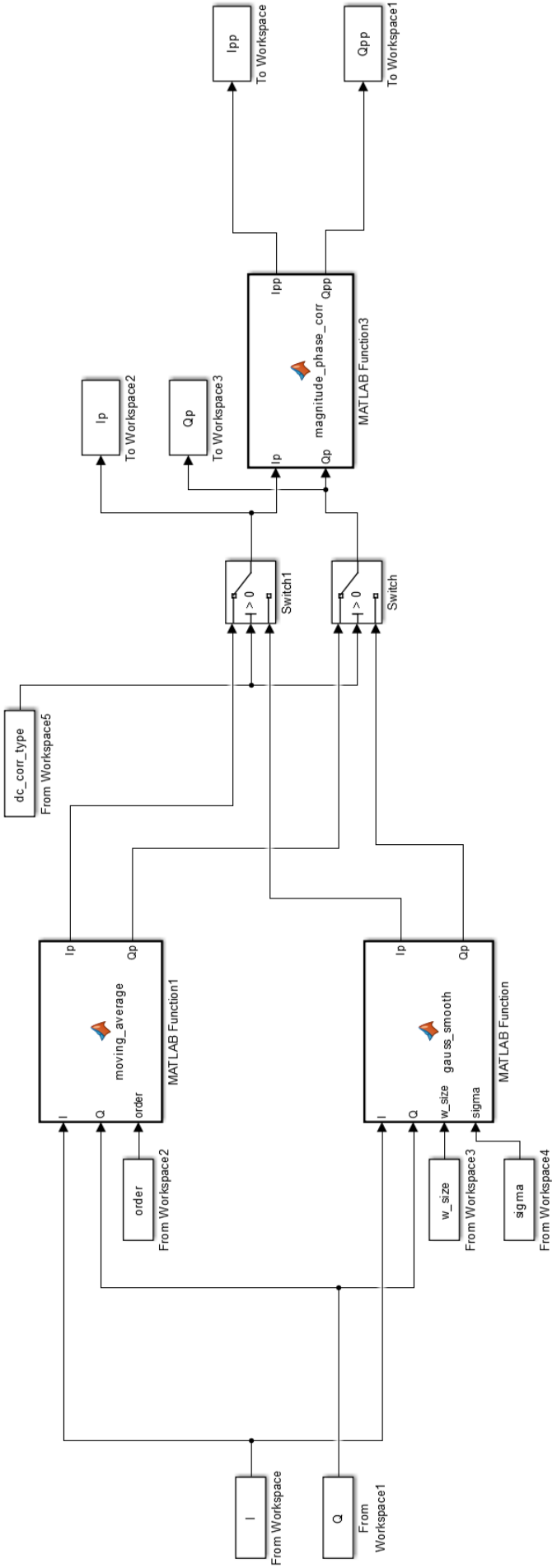

Figure 5.1: *Block diagram of simulation workflow.*

Chapter 6

Hardware Implementation

This chapter explains structure of hardware implemented in Zynq PL section. This includes data flow from PS to AD9361 RF transceiver, including correction algorithms for RF signal quality improvement.

6.1 Data Flow Path

Diagram below shows data path for I/Q samples from Zynq PS system to PL logic.

- *AXI_AD9361* - IP core provided by Analog devices. This block is responsible for interfacing with AD9363 RF transceiver placed on board.
- *FIRDecimator* - down-sampling block with additional filters to block unwanted images created by change of sampling frequency in receive path.
- *FIR Interolator* - up-sampling block with additional filters to block unwanted images created by change of sampling frequency in transmit path.
- *DC Offset Removal* - implementation of dc offset removal algorithm. Depending on test bench Moving Average or Gaussian Filter.
- *I/Q Correction* - realization of phase and magnitude correction in the revived signal.
- *Zynq Processing System* - CPU part of Zynq SoC. Mentioned block is responsible for communication between PL and PS part using AXI interface. Zynq PS section perform task related to data aquisition and interfacing with PC using USB 2.0 interface.

6.2 Movig Average Filter

6.3 Gaussian Filter

6.4 I/Q mismatch compensation

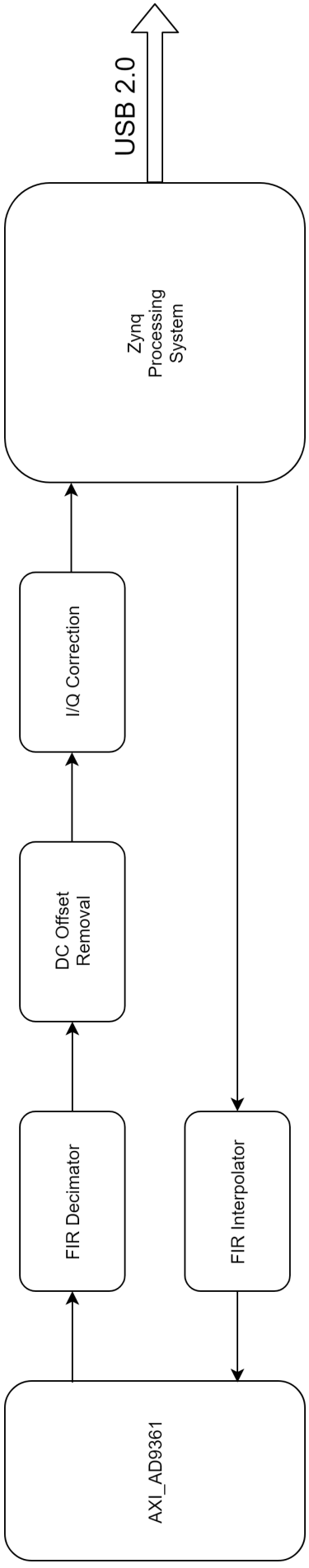

Figure 6.1: *Representation of sine wave in complex plain*

Chapter 7

Performance evaluation

Chapter 8

Conclusions

Bibliography

[1] Stephen H. Hal, Garrett W. Hall, and James A. McCall. *High-Speed Digital System Design - A Handbook of Interconnects Theory and Design Practices*. New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2000.