High Speed Passive Radar Receiver with Application to Digital Television Signals

Weiwei Sun

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

John D. Sahr, Chair

Sumit Roy

Brian Nelson

Program Authorized to Offer Degree:

Electrical Engineering

University of Washington

Abstract

High Speed Passive Radar Receiver with Application to Digital Television Signals

Weiwei Sun

Chair of the Supervisory Committee:
Professor John D. Sahr
Electrical Engineering

In this dissertation we present initial results from the fourth generation receiver for the Manastash Ridge Radar (MRR), which is a distributed passive radar system used for ionospheric physics and engineering studies. This receiver permits simultaneous access to the HF, VHF, and UHF spectrum by sampling at speeds up to 5 billion samples per second on each antennas. This system has large aggregate bandwidth; it can simultaneously collect the entire VHF FM broadcast band as well as several UHF DTV broadcasts. The digitizers have eight-bit precision. The wide bandwidth sampling (oversampling) means that it is possible to accurately sample narrowband signals whose amplitude is much less than the least sampling quantum. Most of the analog signal path is eliminated, yielding excellent dynamic range, and high speed digital signal processing yields low-latency real time operation.

We also present initial data from such a receiver used to support passive bistatic radar experiments. We discuss in detail algorithms to make effective use of the FPGA. For example, the sampler runs 8 or 16 times faster than the FPGA, so initial FPGA processing requires parallel algorithms. In our design the downconverter passband center frequencies and spectral widths are selectable at run time, and can be changed in a few milliseconds. The FPGA operates in fixed point, which presents both opportunities and challenges in managing precision during the signal processing, for networking, and for subsequent signal processing. Data is sent off the receiver via one or more 10 GbE ports. In our current implementation system performance is limited primarily by network bandwidth.

For radar application, the signal to clutter ratio is dominant for radar system performance. DTV signals have a known structure which permits recovery of the original transmitted waveform from imperfect reception. With such a nearly ideal reference signal, we are able to map the multipath and also improve the instantaneous dynamic range to over 100 dB with one-second coherent processing, and thus detect the weak echoes from targets of interest, such as aircraft or ionospheric field-aligned irregularities, and pave the way for AoA estimates or interferometric imaging of these scatterers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ADC: Analog to Digital Converter

ATSC: Advanced Television System Committee. An international, non-profit organization developing standards for digital television.

CIC filter: Cascade-Integrator-Comb filter

DAC: Digital to Analog Converter

dBFS: dB below the full scale power

dBm: an abbreviation for the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW)

DDS: Direct Digital Synthesizer.

In this paper, DDS is referring to Xilinx DDS compiler.

DTV: Digital Television

DVB-T: Digital Video Broadcasting - Terrestrial

ENOB: Effect Number of Bits.

ENOB specifies the resolution of a real ADC circuits which inevitably introduces noise and distortion. It can be expressed as:

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

where SINAD is signal to noise and distortion ratio, Ratio expressed in dB of the RMS full scale signal amplitude to the RMS sum of all other spectral components, including the harmonics except DC.

ERP: Average Radiation Power. It is equal to transmitter power times transmitter gain in the maximum radiation direction.

FFT: Fast Fourier Transform

GbE: Gigabit Ethernet

GSPS: Giga samples per second

I/Q: In phase and Quadrature

KONG: A digital TV channel. It is one of the received DTV channels with high power and clear line of sight.

Mbps: Megabits per second

MRR: Manastash Ridge Radar

MUX: Multiplexer

ROACH: Reconfigurable Open Architecture Computing Hardware. A standalone FPGA processing board.

Rx: short for Receiver

SCR: Signal to Clutter Ratio

In this paper, Clutter in the SCR is the target of interest.

SFDR: Spurious Free Dynamic Range

For the ADC, SFDR is the ratio expressed in dB of the RMS signal amplitude, set at 1dB below full scale, to the RMS value of the highest spectral component (peak spurious spectral component). The peak spurious component may or may not be a harmonic. It may be reported in dB (that is, related to converter –1 dB full scale), or in dBc (that is, related to input signal level).

Tx: short for Transmitter

UDP: User Datagram Protocol. UDP is one of the core members of the communication protocol.

VSB modulation: Vestigial Sideband modulation

ACKNOWLEDGEMENTS

I would like to thank CASPER community for their tutorial and technical advice for ROACH-2 configuration. In particular, I'm grateful to Rurik Primiani from Harvard-Smithsonian Center for Astrophysics; Jack Hickish, David Macmahon and Dan Werthimer from UC Berkeley; John Ford from NRAO; Jason Manley and Marc Welz from SKA; and Glenn Jones from Caltech, for their help on troubleshooting the ADC and receiver network; and to Xilinx, Inc.

I would like to acknowledge The Boeing Company for funding and collaboration on this work. Additionally, I would like to thank my lab mate, Laura Vertatschitsch, for her instructions and advice on ROACH-1 on which our ROACH-2 development is based. It's been a memorable time while we worked together on the project.

DEDICATION

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents with encouragement and financial support throughout the entire doctorate program. My sister Xinxin, has never left my side and are very special.

I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done, especially Jie Wang, Laura Steinkamp for being there for me and helping me with messes again and again without hesitation. Both of you have been my best cheerleaders.

I would like to dedicate my work to my advisor, Prof. John Sahr. He is the best advisor I ever had. He is one of the very few advisors who are willing to advise students with no knowledge in the related research area. He is the one of the very few advisors who advise students with great patience. Whenever I have a question on this project, he can always instruct and direct me to the solutions. It's his motivation and encouragement that drive my interest in radar system deeper and deeper. Without his help, I might not survive the PhD program; without his advising, I might not learn and achieve what I have right now. I'm so fortunate to have John as my advisor in my life.

I would give special thanks to my Lab mate Laura Vertatschitsch for helping me get started on this project and instructions on DSP fundamentals.

My thanks to my labmate Marcos Inoñán. He made the lab lovable and warm.

I would like to thank the committee, Prof. Sumit Roy, Prof. Brian Nelson. Their excitement and willingness to provide feedback made the completion of this research an enjoyable experience.

Chapter 1. INTRODUCTION

1.1 INTRODUCTION TO RADAR AND PASSIVE RADAR

Radar is a technique using radio waves to detect a variety of targets, from well underground (glacier and ground penetrating) to the moons of Saturn. Based on the geometry, radar can be classified as monostatic radar or bistatic radar. In bistatic radar, the transmitter and receiver are separated by a distance that is comparable to the expected target. Conversely, monostatic radar is a radar in which the transmitter and receiver are collocated. Passive radar is inherently a special case of bistatic radar. It uses non-cooperative sources of illumination in the environment as its transmitter, instead of a dedicated transmitter.

Radar technology is now just about one century old. Among the oldest radar examples is the ionosonde built by Breit and Tuve in the mid 1920s. However, the first radar is often attributed to Watson-Watt in 1935, recognizable as an air surveillance system. Interestingly, Watson-Watt's Daventry experiment was a passive bistatic radar, with which we are presently concerned. Passive radar detects and tracks targets with reflections from non-cooperative transmitters. The transmitters can be commercial broadcast and communication signals, such as FM broadcast, TV broadcast, GPS, WiFi, mobile signals, etc. Passive radar has been widely used to observe geophysical or aerospace targets. A variety of commercial broadcast, communications, and navigation transmitters have been studied for their use in passive radar systems [1-12].

In general, passive radars lack precise knowledge or control of the transmitter waveforms. However, the waveforms and 100% duty cycle of modern communications systems yield high average power, large processing gains (40-70 dB) and good range resolution through correlation processing for typical applications. This correlation processing, generalized to estimation of the cross-ambiguity function [5, 13, 14] permits estimation of slant range and Doppler information for deep fluctuating targets (our primary interest). High spatial resolution can be achieved by observing multiple transmitters simultaneously, using TDOA (time difference of arrival) processing, by interferometry [5, 15, 16], and other techniques, such as Frequency Domain Interferometry [17].

VHF and low UHF broadcast transmitters are useful for radar studies of the ionosphere because the meter-scale wavelength interacts effectively with meter scale turbulence [18]. These

illumination sources are also useful for detection of aircraft [7, 11] and spacecraft [19]. Although there are challenges associated with passive radar, there are also interesting benefits: there is no transmitter licensing or expense to bear; they are inherently safe; they cause little or no radio frequency interference; and as receive-only systems they are inherently stealthy. Passive bistatic radars depend heavily upon computation; fortunately computing power has increased dramatically in the past two decades.

The conventional radar receiver adopts a super-heterodyne architecture, as shown in Figure 1.1. The RF signal is received on an antenna, then passes through an image reject filter. The filtered signal later is mixed down to intermedia frequency with the analog mixer, and filtered with an anti-aliasing filter. The signal then is split into two streams and passes through two 90-degree phase shifted mixers which converts the signal to baseband in phase/quadrature (I/Q) signal. The baseband I/Q signal then is digitized with a low rate Analog to Digital Converter (ADC). In a slightly different architecture, the intermediate frequency signal is digitized with one medium rate ADC directly. The baseband down-conversion is implemented in digital form, rather than analog mixer. Because of the analog components used in both architectures, the performance of this type of radar depends upon the linearity of these analog components. A large literature on receiver design is availables, see [20, 21]

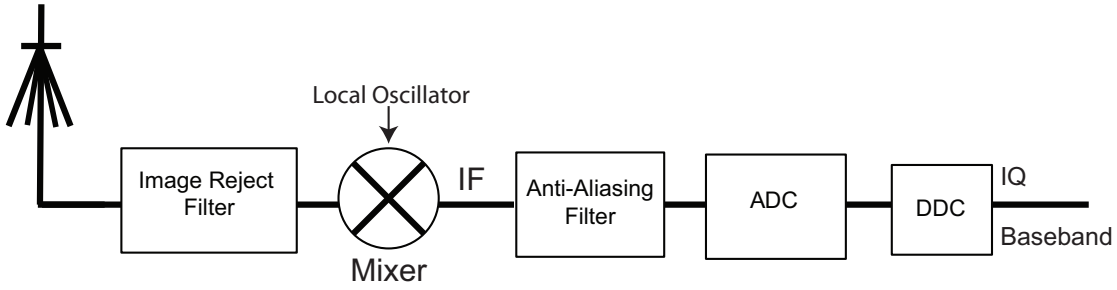


Figure 1.1. An example radar receiver that receives signals on an antenna, mixes the data down to an intermediate frequency with an image reject filter and analog mixer pair, applies an anti-aliasing filter and samples the signal at IF with the Analog-to-Digital converter (ADC), and finally applies a digital downconverter to mix the sampled IF signal down to IQ and minimize the data rate. Passive radars often do not require signal amplification, but other radar receivers may include a low-noise amplifier (LNA) after the antenna [22]. Radar receivers commonly use two or more frequency translation stages.

1.2 Overview of Passive Radar System

In this chapter, we will discuss the existing passive radar systems in the world that scientists and engineers explored by using different transmitter sources and different methods to achieve better detection performance. The systems in section 1.2.1 are based on FM, DAB and analog television signals, and the system in section 1.2.2 are based on DTV signals. At the end, we will introduce the history of our lab, RRSL lab in the department of Electrical Engineering at University of Washington, which has developed five generations of passive radar systems. The RRSL lab has decades of experience in radar system design. The research area involves ionospheric irregularity study, using the radar system that built in the lab to observe plasma turbulence in the lower ionosphere [23, 24]. In recent years, we have been taking on a project funded by The Boeing Company that builds a new generation passive radar system to detect small targets of interest.

1.2.1 *Early Passive Radar Studies of Using FM, DAB, Analog Television Broadcasts*

A few passive radar systems in the world have been developed with excellent real-time performance with traditional analog broadcasts, such as FM, DSB, analog TV [3, 5, 6, 11, 12, 25-28]. The systems typically have two channels, a reference channel for clear reception of the transmitted signal, and one or more surveillance channels for good reception of echoes from the target. The reference channel antenna has a high gain in the direction of the transmitter and a null in other directions to suppress multi-path. Signal processing methods, such as auto-regressive processing [26] and the CMA algorithm [29], are used to further extract the direct path. The surveillance channel antenna usually has a deep null in the direction of transmitter to allow high SQNR (Signal to Quantization Noise Ratio) at the ADC and suppress direct signal interference. A variety of adaptive filtering algorithms are adopted to suppress the direct path and multi-path clutter. The newly emerged commercial-off-shelf, inexpensive USRP (Universal Software Radio Peripheral [30]) provides options of ADC, DAC daughter board and customizable signal processing chains in FPGA.

Phased array beam forming and interferometry are also implemented for Angle of Arrival (AoA) estimate. Multiple transmitters are explored to improve the spatial resolution. Most systems use Kalman filters for tracking, and cell averaging constant false alarm rate (CA-CFAR) processor

[31] for improving the detection of low SNR target and reducing false alarms caused by noise and interference.

1.2.2 *DTV-Based Passive Radar*

With recent advances of solid-state electronics, we have entered an era when true digital communications are ubiquitous. Compared to the earlier communication with analog circuits, digital communications make more efficient use of the spectrum, and can add error correction and encryption, and thus could better overcome interference issues. In the United State, television broadcast has recently made the transition from analog NTSC to digital 8VSB [32]. Digital television broadcasts (terrestrial) have bandwidth between 6MHz and 10MHz depending on the standard in different counties, which is much larger than the 100 kHz bandwidth of FM. For radar applications, DTV provides much better range resolution than FM. Therefore, the DTV signal has attracted significant attention in waveform studies and related passive radar system development [9, 33-35]. DTV broadcasts in different counties adopt different modulation scheme. In Europe, DTV signals use the DVB-T standard, employing orthogonal frequency division multiplexing (OFDM). The signals are constructed based on a standard [36] which includes data frames, guard intervals, and pilot information that is inserted for use by a DVB-T receiver for synchronization and for channel estimation. However, the structure of the DVB-T signal give rise to peaks in the radar ambiguity function, which could mask returns from real target. It is widely adopted in Europe, Australia and some other countries.

DTV signals in North America, however, adopted a different modulation scheme -- Vestigial Sideband Modulation (8-VSB) which has a significantly different structure from the ATSC DTV standard [37]. Unlike the extensive studies of OFDM modulated DVB-T waveform, there have been few studies of 8VSB for passive radar [38, 39], however, these studies reveal excellent performance.

1.2.3 *History of RRSL Lab*

The UW Radar Remote Sensing Laboratory (RRSL) is where we have developed the new receiver. Our lab has built several passive radars, named Manastash Ridge Radar (MRR), with excellent system performance. They have made significant contributions to the study of the ionospheric irregularities. In this section, we will briefly describe the architecture of each MRR generation.

The first generation MRR receiver [5] was built in 1997 by John D. Sahr and Frank D. Lind. It was built to study the upper ionosphere with commercial FM broadcasts near 100MHz. The receiver had a particularly simple direct conversion topology, as shown in Figure 1.2. The signal received by the antenna is directly converted to baseband with a single I/Q mixer with local oscillator set to the center frequency of the selected radio station. Because there is no transmitter near the receiver, the image rejection filter is eliminated. The I/Q mixer is followed by low pass filtering and additional amplifiers, at which point the I/Q signals are sampled. There were two receivers, one located at University of Washington, the other at Manastash Ridge in which the direct FM signal is blocked by the Cascade Mountains. The two receivers were synchronized with Global Positioning System (GPS) receivers and data from these two receivers were streamed to a centralized signal processing computer for analysis.

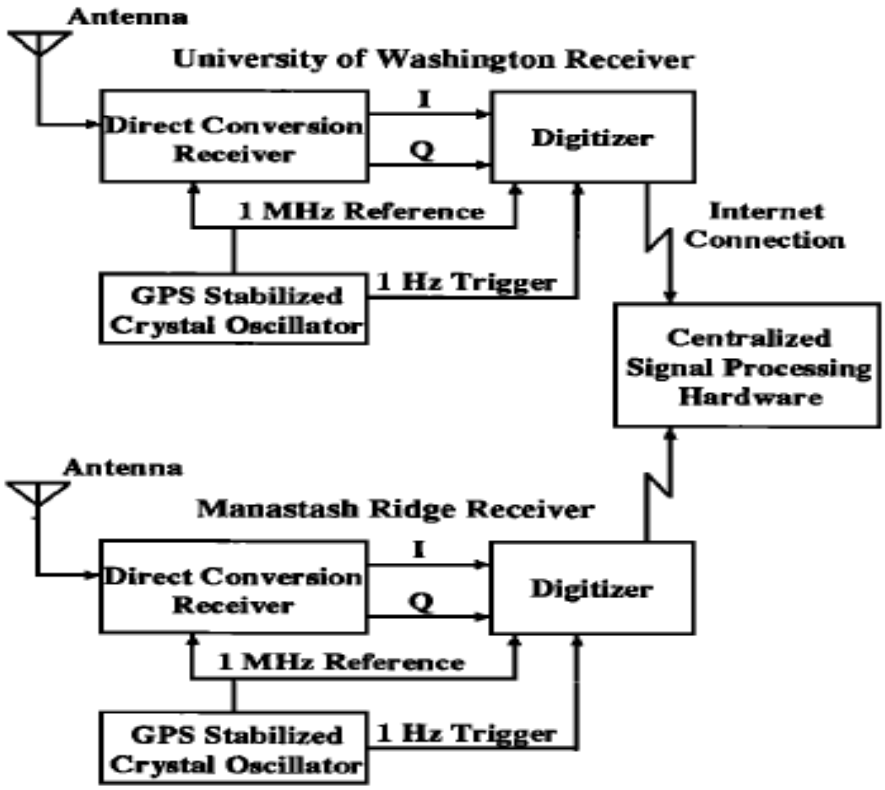


Figure 1.2. Block diagram of the first generation MRR receiver [5]. The two receivers exchange data via a true, fast internet connection. The sampling and mixing synchronization is provided by Global Positioning System receivers.

The second generation of the MRR, such as that of the Intercepted Signals for Ionospheric Science (ISIS) receiver [40, 41] had greatly improved performance capabilities. The ADC can operate at 72 or 105 Mega samples per second (MSPS). Figure 1.3 shows the process for the ISIS receiver to observe both DTV signals and FM signals. The top path shows DTV processing, which follows the typical radar architecture discussed earlier. The FM path below shows the mixer is eliminated entirely. For FM signal reception, the ADC operates at 72 Mega Samples per second (MSPS), and therefore, FM signals that range from 88 MHz to 108 MHz is intentionally aliased to lower frequency.

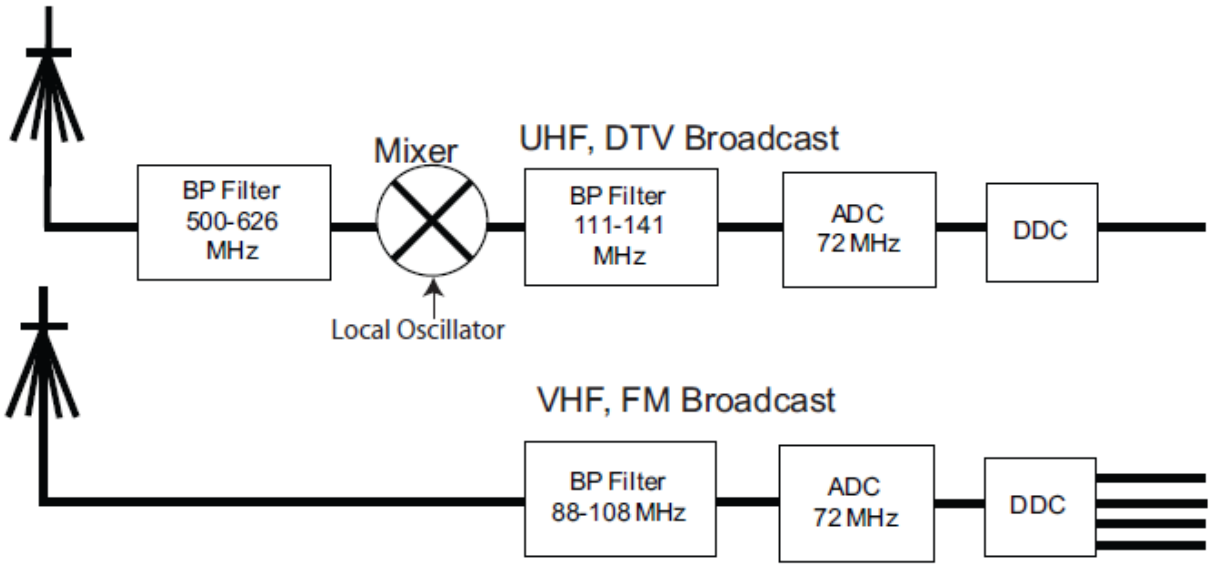


Figure 1.3. The second generation of MRR radar: the ISIS passive radar system [40, 41]. The bottom path is for FM reception with a 72 MHz ADC which intentionally alias FM signal to lower frequency. The top path is for DTV reception with an additional analog mixer which converts DTV signals to an intermediate frequency before digitizing.

The first and second generation receivers [5, 16, 40, 42-47] use a once-aliased direct RF sampler following a VHF band-pass (Nyquist) filter. These receivers worked well for passive radar using commercial FM broadcasts. However, they were insufficient for Digital TV signals at 600 MHz. First, they required analog down conversion channels to reach 600 MHz; second, the digital receiver hardware could not easily pass the 6 MHz bandwidth of a DTV channel. This limitation led us to develop the third and fourth generation receivers [22, 48, 49] to address this challenge

with much higher sample rates (1 – 5 Giga samples per second (GSPS)). The new receiver discussed in this dissertation is the fourth generation receiver, but they are both adapted from the Reconfigurable Open Architecture Computing Hardware (ROACH) board developed by the Collaboration for Astronomy Signal Processing and Electronics Research (CASPER) [50]. Each ROACH board contains a large FPGA, and one or two ADC cards may be attached to the board. In practice, we used two dual 1.6 GSPS samplers for the third generation ROACH-1 (with Xilinx™ Virtex 5™ FPGA) and two dual 2.5 GSPS samplers for the fourth generation ROACH-2 (with Xilinx™ Virtex 6™ FPGA).

Due to the high speed samplers, both the third and fourth generation receivers use direct sampling architecture. The analog mixers are completely eliminated. With an appropriate low pass or bandpass filter, the signal from the receiver is directly digitized, and followed by direct down conversion which is implemented in the FPGA.

Figure 1.4 shows the block diagram of ROACH-1. ROACH-1 has two ADC boards (KATADC [51]). Each of them can operate with two channels up to 1.5 GSPS or in single channel mode up to 3 GSPS. It can capture FM and DTV broadcasting signals simultaneously without duplicated signal paths. One of the few analog components is a band-pass filter that isolates a portion of the DTV spectrum from 500-626 MHz. The outputs of these filters connect directly to the two KATADC boards on the ROACH. Before the digitizer, the KATADC board has a fixed front end amplifier with passband of 50 MHz - 6.0 GHz and gain of 20 dB followed by a software-controlled attenuator that varies from 0 to -31.5 dB. Vertatschitsch has made many contributions for the ROACH-1 development [22]. She successfully demonstrated the following modes:

(1) two antennas, 1.6 GSPS, two 6 MHz spectral windows (2 DTV stations per antenna) 8-bit IQ Output;
(2) Four antennas, 800 MSPS, one 6 MHz spectral window, 8-bit I/Q Output;
(3) Two antennas, 1.6 GSamp/s, two 60 MHz spectral windows, 8-bit I/Q Output;

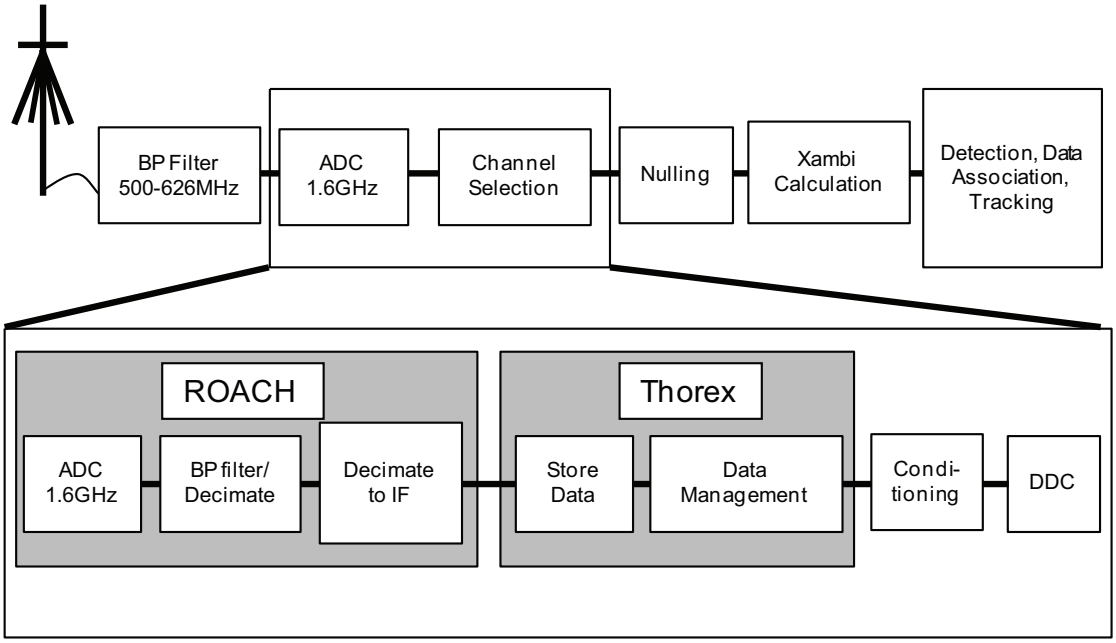


Figure 1.4. Block diagram of the ROACH-1 receiver architecture [22]. ROACH is the receiver board. It oversamples DTV signal, then performs a digital down conversion and decimation to reduce date rate. Thorex is a computer which acquires data from ROACH and performs post-data processing for different applications.

The digitizer provided in ROACH-2 board supports sampling rate up to 5 GSPS for single input, compared to the single 1.6 GSPS digitizer equipped in ROACH-1 board. ROACH-1 supports up to four 10-Gega-bit Ethernet (10GbE) ports for data transmission, where ROACH-2 could support up to eight 10GbE ports. The two ROACH boards adopted different FPGA chips as well: a Xilinx Virtex 5 for ROACH-1 and Virtex 6 for ROACH-2. Table 1.1 lists the FPGA resources for Virtex 5 and Virtex 6. The Virtex 6 FPGA in ROACH-2 has more CLBs, RMA blocks, DSP48E slices and more clock options than the Virtex 5 FPA equipped by ROACH-1. Loosely speaking, the ROACH-2 FPGA has four times the compute capacity as the ROACH-1 FPGA. Later we will show that the ROACH-2 FPGA perform about 0.5 TOPS (Tera Operations Per Second).

Table 1.1. FPGA resource of Virtex 6 (ROACH-2) vs. Virtex 5 (ROACH-1)

		ROACH-1	ROACH-2
Model		XC5VSX95T	XC6VSX475T
Configurable logic blocks (CLB)	**Slices**	14,720	74,400
	Max Distributed RAM (Kb)	1,520	7,640
Block RAM Blocks	**18 Kb**	488	2,128
	36 Kb	244	1,064
	Max (Kb)	8,784	38,304
DSP48E Slices		640	2,016
Ethernet MACs		4	4
Interface Blocks for PCI Express		1	2
Total I/O Banks		19	21

1.3 Review of Receiver Architectures

1.3.1 *Direct Sampling Receiver*

There are two types of high speed receivers, the conventional heterodyne hardware receiver and the direct RF-sampling receiver. The conventional receiver generally has a medium speed sampler and a few stages of analog mixers and filters for down-conversion. The ADC is usually 12-16 bits resolution, which provides large dynamic range. However, there are a few drawbacks that limit its application and performance. The analog components introduce inter-modulation products and harmonics due to their residual nonlinearities. The two IF mixers should have exactly 90-degree phase shift to prevent I/Q channel imbalance. The bandwidth and dynamic range is pre-determined by the sampling rate of ADC; you cannot detect widely separated frequencies at the same time with a single hardware path, nor detect a different frequency without replacing or tuning hardware [52].

With the improvements in ADCs and computation power, receiver systems have seen a change from the widely adopted heterodyne architecture to a direct RF-sampling approach, which is also called software radio or software defined radio. In an ideal RF-sampling architecture invented by Mitola [53], the high speed sampler digitizes a large chunk of frequency spectrum directly at RF and hands it off to a signal processor to dissect the available information. It takes what has

traditionally been handled by analog processing (mixers, local oscillators and their attendant filters and amplifiers) into the digital domain.

Its main advantage is greater programmability, and the relative ease with which any number of DSP channelizers can be added to receive greater numbers of channels concurrently. Its main limitation stems from the sample rate and dynamic range of practical ADCs. However, many of the interesting wireless bands, such as WLAN, GSM, GPS, Bluetooth, use carrier frequencies well above 1 GHz.

In wireless communication, it is usually thought wasteful to sample so fast as to cover the bandwidth up to 5 GHz. Therefore, most modern communication receivers use software-defined architecture, adding analog hardware at the front end followed with a medium rate sampler to meet the performance requirements [54]. Some adopted an RF analog passband filter followed by a low rate sampler to take the advantage of intentional aliasing [55, 56]. Some adopted analog decimation at the front end [57-59]. Some used integrate-and-dump sampling, followed by analog downsampling and filtering [60-62]. The Universal Software Radio Peripheral (USRP) [30], together with the open source GNU Radio Software [63], have been used to implement sophisticated, yet low cost, software-defined radios for their options of tunable analog mixers at the front end. Unlike the fixed mixer for a particular frequency, the tunable mixer and programmable FPGA offer the opportunity for the same receiver to receive and demodulate different frequency bands. One of the advanced USRP models, USRP X300 [64], covers frequencies from 10 MHz to 6 GHz with 160 MHz instantaneous bandwidth and 16-bit ADC sensitivity. However, the insertion analog mixers, prefixed or tunable, still has the same drawback as the heterodyne hardware receiver, such as the residual nonlinearities, I/Q imbalance, and also imposes limitations on reception of widely separated frequency bands.

The new receiver described in this work adopts the ideal RF-sampling architecture. It features two high speed ADCs (2.5 or 5 GHz) for direct sampling, and a powerful Xilinx Virtex 6 FPGA for high speed signal processing. It has superior linearity and large instantaneous bandwidth due to the elimination of analog down-conversion stages; it provides large processing gain, which enables weak target detection with the ADC's 8-bit resolution. Similar receiver structure can significantly improve passive radar capability.

1.3.2 *Software Defined Radio*

Among the software-defined receivers, software-defined radio (SDR) is a well-known and widely used radio communication system. The components that have been conventionally implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system.

Most receivers use a variable-frequency oscillator, mixer, and filter to tune the desired signal to a common intermediate frequency or baseband, where it is then sampled by the analog-to-digital converter. However, in some applications it is not necessary to tune the signal to an intermediate frequency and the radio frequency signal is directly sampled by the analog-to-digital converter (after amplification).

There are technology limits on achievable RF performances, the choice of architecture which depends on the available technology e.g. ADC performance, semiconductor technology, software reliability (or the lack thereof) which may define overall radio reliability. USRP (Universal Software Radio Peripheral) is the commercial SDR. They are designed and sold by Ettus Research, LLC and its parent company, National Instruments.

One of its most advanced model is USRP X310. Figure 1.5 shows the block diagram of USRP X310. The hardware architecture combines two extended bandwidth daughterboard slots covering DC – 6 GHz with up to 120 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, Dual 1/10 GigE). It has two ADCs and two DACs followed by a digital processor. Table 1.2 shows the FPGA model in USRP N210, X300 and X310. In model X310, the processor is equipped with Xilinx Kintex 7-410T with 406k Logic Cells, 28 Mkb block memory, 1540 Multipliers and 200 MHz clock rate. The ADC has a maximum sampling rate at 200 MSamp/sec and a resolution of 14 bits. Table 1.3 compares FPGAs for USRP X310 and ROACH-2. The advanced X310 model does not have as many slices and dedicated DSP blocks as ROACH-2.

Table 1.4 compares the two products on the system level. We can notice that, beside the FPGA slice number, the FGPGA clock and the maximum speed of the ADC also has a large difference: ROACH-2 board runs much faster than USRP X310.

Table 1.2. The FPGA used in USRP models N210, X300 and X310

	USRP N210	**USRP X300**	**USRP X310**
FPGA	Spartan3 XC3SD3400A	Kintex 7-325T	Kintex 7-410T
Logic Cells	53k	321k	406k
Memory	252kb	16,020 Kb	28,620 Kb
Multipliers	126	840	1540
Clock Rate	100 MHz	200 MHz	200 MHz
Streaming Bandwidth per Channel (16-bit)	25 MS/s	200 MS/s	200 MS/s

Table 1.3. FPGA resource of Virtex 6 (ROACH-2) vs. Kintex 7-410T (USPR X310)

		USRP X310	**ROACH-2**
Model		Kintex 7-410T	XC6VSX475T
Logic Cells		406,720	476,160
Configurable logic blocks (CLB)	**Slices**	63,550	74,400
	Max Distributed RAM (Kb)	5,663	7,640
Block RAM Blocks	**18 Kb**	-	2,128
	36 Kb	795	1,064
	Max (Kb)	28,620	38,304
DSP48E Slices		1,540	2,016
Interface Blocks for PCI Express		1	2
GTX Transceivers		16	36

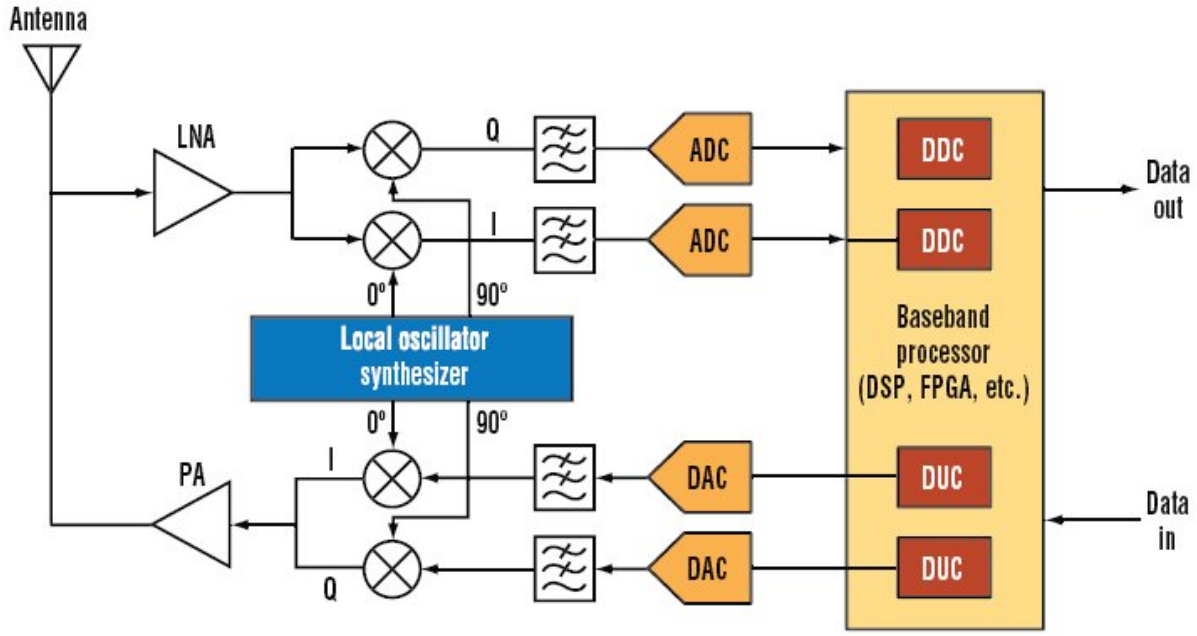


Figure 1.5. Block diagram of the USRP model X310. It has median speed ADC and DAC up to 200 MSPS, and analog mixer and filters are required at the front for high frequency signals.

It's relatively inexpensive (< $2.5k with a down-converter) compared to ROACH-1 ($24k), light weight, equipped with 2 transmitters, and has a friendly user interface from integration with National Instruments LabVIEW or GNUradio. However, the ADC of USRP is relatively slow (i.e. 200 MSamples/sec), so additional electronic elements, such as analog mixer, are required for signals in the UHF band. On the other hand, ROACH-1 and ROACH-2 have much faster samplers which avoid non-linearity that those electronic elements bring, and furthermore, no duplicate paths are required to capture VHF and UHF signals at the same time. Another minor point is the ROACH-2 support up to eight 10GbE ports and one 1GbE port, while the USRP can only support only a single 1 1GbE in most of its models.

Table 1.4. Comparison of USRP (X310) and ROACH-2

		USRP X310	**ROACH-2**
Model		Kintex 7-410T	XC6VSX475T
Logic Cells		406,720	476,160
ADC	**Resolution**	14 bits	8 bits
	Sampling Rate (Max.)	200 MSPS	5 GSPS
FPGA	**model**	Kintex 7-410T	Virtex 6
	Dedicated DSP slices	1540	2016
	Clock Rate	200 MHz	400 MHz
10GbE interfaces		1	8
Cost		< $4K	$ 24K

1.4 FPGA DESIGN, SIMULATION, COMPILATION

1.4.1 *Design Environment*

For ROACH-1 and ROACH-2, we create the FPGA design in Matlab Simulink. Simulink® is a block diagram environment for multi-domain simulation and Model-Based Design. Figure 1.6 shows an example of Simulink GUI. It supports simulation, automatic code generation, and continuous test and verification of embedded systems.

Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. All the functions are represented with blocks, instead of VHDL code. The Simulink will later convert the block design into VHDL code, which will then be compiled into the bit stream that will configure the FPGA.

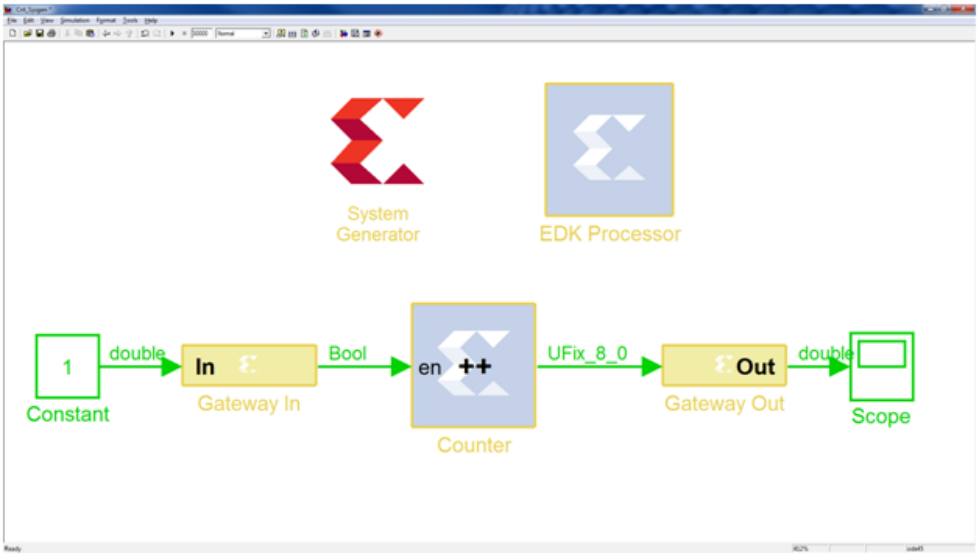


Figure 1.6. An example of Simulink design. Gray blocks are Xilinx blocks. Green blocks are Simulink blocks for generating or displaying various waveform in simulation. Gateway in/out are interface to communicate with Xilinx blocks. System Generator token is required to apply the correct platform for the gray Xilinx blocks.

With Simulink built-in blocksets, we can implement a simulation of the design. To integrate with Xilinx Product, we need the Xilinx System Generator and the Xilinx blockset which are provided by Xilinx and can be linked from Simulink. The System Generator token, shown as the red block in Figure 1.6 above, needs to be added to the design. The System Generator token provides users to select specific target workflow and platform.

The Xilinx Blockset contains over 90 DSP blocks, ranging from simple adders, multipliers etc. to complex blocks such as Forward Error Correction blocks, FFTs, filters and memories, etc. As far as the rest of the modeling design process is concerned, the Xilinx block works quite similar to other Simulink blocks, i.e., users drag and drop various blocks into the Simulink environment in order to design the overall system. Users can then use MATLAB scopes and other sinks as well as elements of Sources library to check the design results. We were able to make the receiver work by using these high level tools rather than wrting VHDL code. While a pure VHDL solution might have better performance, the Simulink environment permits testing algorithm conveniently and rapidly, and guarantees synchronization of the desired algorithm and the resultant FPGA code.

These blocks leverage the Xilinx IP core generators to deliver optimized results for the selected device. The System Generator also includes the mcode and Black Box blocks, which can be used to integrate mcode and HDL codes, respectively, directly into the Simulink design environment. Since System Generator is already part of Xilinx ISE™ or Vivado™ Design Suite, no additional synthesis tools are required and the users can generate the bitstream directly from within the Simulink environment. All of the downstream FPGA implementation steps including synthesis and place and route are automatically performed to generate an FPGA programming file for the expected Xilinx platform. For people who work closely with similar platform or project, the Xilinx System Generator for DSP (Getting Started Guide) is highly recommended [65]. It gives detailed descriptions from System generator, Xilinx blocks, installation, fix-point operations, filter design, using memories, to advanced topics, such as creating black blocks with MCode, multi-rate system, etc.

ROACH-2 is developed by selecting the most powerful hardware, such as high speed ADC, Xilinx Virtex 6 series FPGA, large RAM. The customized board has numerous pins for connections and calibration, which makes write HDL codes for the interfaces a big pain, especially for uses who don't have knowledge of embedded system programming, but hope to program the system for scientific research. As a CASPER engineer once said, 'if you want to write your own HDL codes from scratch, it will be very very bloody.' Fortunately, the CASPER community has pre-programmed the interfaces and gateway, such as ADC, 10GbE, FPGA RAM, and integrated them to Simulink blocks called yellow blocks. These yellow blocks are in the library of *CASPER_XPS System Blockset*. Users can directly use the gateway blocks in Simulink, and adjust parameters as needed. These yellow blocks are necessary for almost every design for ROACH-2 board. In chapter 3, we will discuss some of those yellow blocks.

1.4.2 *Compilation*

After a design is created in Simulink, simulation is first performed before compilation to make sure the logic and calculation results are as expected. As Simulink is integrated with MATLAB®, we can incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

After a thorough simulation analysis is done, the design is compiled to produce a borph-recognizable file which can run on the ROACH-2 board. Here, the BORPH (http://www.borph.org)

is an operating system designed for FPGA-based reconfigurable computers The compilation flow is shown in Figure 1.7.

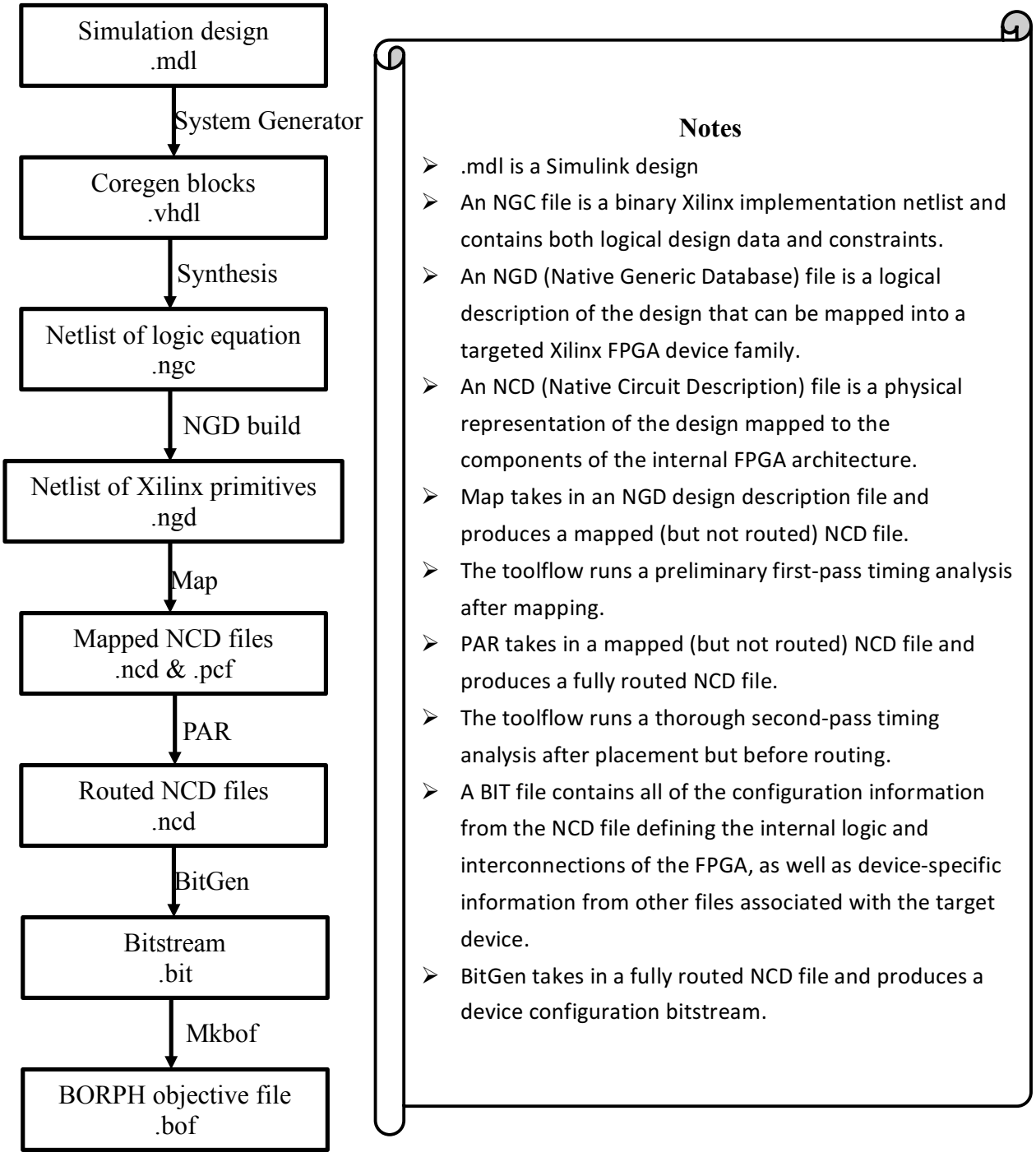


Figure 1.7. Compilation flow of Simulink design.

Because our design runs above 300MHz on ROACH-2, and Simulink is insufficient for speed optimization, compilation in Simulink is unable to meet timing requirements and usually fails. Planahead software provided by Xilinx is used to solve this problem. One solution is to manually constrain placement of primitives on the FPGA fabric (guidance can be found in [66]). If it still cannot meet timing, a closer analysis of the resource mapping on the device produced by PlanAhead is necessary. The resource mapping gives information of which critical paths failed and where they are located. Then we can analyze why they failed and make changes in the Simulation to relax the timing of signal propagating from one place to another.

1.4.3 *Python and Corr Library*

As a BORPH file is running on ROACH-2 board, there are multiple ways to interact with the board, such as TinySHell, BORPH or Corr. [67]. In our work, we mainly use Corr. Corr is a Python library that implements the Karoo Array Telescope Control Protocol (KATCP). It eliminates the need for everyone to write their own FPGA control code, and its environment – python is a programming language that can run on many operation systems and also interact with other host commands and resources. For example, with our python codes, we can interact with ROACH-2 board and invoke other C files or system command at the same time to delivery real time monitoring and signal processing results.

1.5 PROJECT GOAL AND ORGANIZATION

Our primary engineering goal is to develop a passive bistatic radar receiver which enable detect a small target by achieving 100 dB dynamic range. Direct sampling receiver architecture with a high speed ADC and FPGA is chosen for the capability of multiply frequency channels across HF and UHF spectrum and the processing gain achieved from downsampling. Conventional ambiguity function is used to further increase processing gain and thus increase the visibility of weak targets.

The challenges lie in two aspects. First, FPGA timing is often difficult to close. The receiver with the direct sampling architecture requires designing FPGA at unusually high speed. Furthermore, multiply frequency channels use a large amount of FPGA resources, and down sampling will significantly grow the signal bit width, both of which make timing closure even

difficult. Finally, the tradeoff between signal precision and timing closure has also to be made. Limited signal precision poses the issue of excess truncation noise and overflow, which need to be taken care of as well.

The second challenge is the high amplitude of the direct path and ground clutter at our antenna. It contributes large numerical noise to the ambiguity function, and the small target of interest is below the computational noise and thus missed. The digital demodulation of DTV signal is introduced to suppress the direct path and ground clutter.

We are able to overcome these two major challenges and other related problems, and deliver a passive radar system with great capability. In this dissertation, I will introduce the new passive radar system and demonstrate its detection capability of small targets.

The dissertation is organized as follows. Chapter 2 provides information of the system hardware. Chapter 3 presents the FPGA designs including the signal flow, Xilinx blocks implemented and simulation results. Chapter 4 presents the DTV digital demodulation techniques whose purpose is to achieve sufficient dynamic range for weak target detection.

Chapter 2. SYSTEM ARCHITECTURE

In this chapter, we will give an overview of the system and the hardware composition. We will also discuss some system level topics, such as radar detection requirement based on radar equation, noise sources and analysis, frequency distributions of signals presented on our antenna and spectrum analysis.

2.1 DETECTION REQUIREMENT

Passive radars are in general not limited by noise power, but instead by the signal-to-clutter (SCR) ratio. The SCR compares the received power from a target to the received power of interference. Usually the interference directly comes from transmitter (direct path) or come from the strong echoes of clutters (multi-path), such as buildings, bridges and the ground. The received power P_0 of the direct path can be defined using the Friis free space transmission equation:

$$P_0 = P_t \frac{\lambda^2}{(4\pi)^2} \frac{G_t G_r}{R_{tr}^2} \tag{2.1}$$

where P_t is the transmitter power; λ is the wavelength of the transmitted waveform; G_t is the gain of the transmit antenna in the direction of the receiver; G_r is the gain of the receive antenna in the direction of the transmitter; R_{tr} is the distance between transmitter and receiver. This describes how the transmitted power spreads as it travels through free space. This description, however, does not account for a variety of propagation phenomena such as ducting, or Lloyd's mirror. The latter considers the destructive and constructive interference of this direct path signal reflecting off of the ground. Also not considered here are system losses such as transmitter impedance mismatch, transmitter illumination loss, etc [68].

Here we will calculate SCR based on the geometry of the receiver and transmitter location, and then derive the requirement for our new receiver. The requirement is very important to the system design, as it is the goal of building this high speed receiver, and also the motivation behind this dissertation.

First we select TV station KONG as the source of the DTV signal based on the frequency analysis of all DTV signals around. The details of the frequency analysis are described in section 2.7. The receiver is located on the roof of Seig Hall, a 4-story building on the campus of University

of Washington. The distance between transmitter tower and our receiver antenna R_{tr} is about 4.2 km. Based on Equation 2.1, we can calculate the power received at the UW site from the DTV transmitter KONG (on Queen Anne Hill) P_0:

$$P_0 = P_t \frac{\lambda^2}{(4\pi)^2} \frac{G_t G_r}{R_{tr}^2} = 1000\text{kW} \frac{(0.52\text{m})^2}{(4\pi)^2} \frac{1}{(4.2\text{km})^2} = -10.1\ \text{dBm} \qquad (2.2)$$

Here we assume the transmitter and receiver antenna are both isotropic, (G_r=1); and from FCC data, the effective radiated power (ERP = P_tG_t) is 1000 kW, the central frequency of DTV channel is 575MHz.

The power received due to the signal scattered from a target, P_s, can be shown as:

$$P_s = P_t \frac{\lambda^2 \sigma}{(4\pi)^3} \frac{G_t G_r}{R_{ts}^2 R_{sr}^2} \qquad (2.3)$$

where σ is the bistatic radar cross section (RCS) of the target, G_t is the gains of transmitter and receiver antenna at the direction of the target. If we choose an arbitrary 0 dBsm target seen at a distance of 5 km from the receiver, we can expect the signal power scattered from the target:

$$\begin{aligned} P_s = P_t \frac{\lambda^2 \sigma}{(4\pi)^3} \frac{G_t G_r}{R_{ts}^2 R_{sr}^2} &= 1000\text{kW} \frac{(0.52\text{m})^2 \cdot 1}{(4\pi)^3} \frac{1}{(9.2\text{km})^2 (5\text{km})^2} \\ &= -101.9\ \text{dBm} \end{aligned} \qquad (2.4)$$

Here we have used R_{sr} = 5 *km*, and estimated a worst case $R_{ts} = R_{sr}$ + 5 km = 9.2 km, forming a straight line from transmitter to receiver to scatterer. Combine Eq. 2.2 and Eq. 2.4, and assume isotropic antenna, we can define the SCR here as :

$$SCR = \frac{P_s}{P_0} = \frac{\sigma}{4\pi} \frac{R_{tr}^2}{R_{ts}^2 R_{sr}^2} = -101.9\text{dBm} - (-10.1\text{dBm}) = -90.8\ \text{dB} \qquad (2.5)$$

For a 0 dBsm target at 5 km away from receiver, we would require the receiver to achieve 100 dB dynamic range, or a noise floor that are 100 dB below the direct path.

We will see later in Chapter 3 the correlation of signal after the processing in FPGA can provide 52 dB dynamic range. The advanced signal processing algorithms based on Digital TV

demodulation and adaptive filtering algorithm in Chapter 4 will increase the dynamic range by anther 50 dB to meet the system requirement.

2.2 System Architecture Overview

The new receiver is an extension of earlier work in the Radar Remote Sensing Laboratory at the University of Washington. More history about RRLS lab can be found in Chapter 1. The new receiver developed in this work adopted the ideal RF-sampling architecture. It features two high speed ADCs (2.5 or 5 GHz) for direct sampling, and a powerful Xilinx Virtex 6 FPGA for high speed signal processing. It can directly capture any signal of frequency up to 2.5 GHz without aliasing. It significantly expands the available signal sources for passive radar application, including but not limited to FM broadcast, TV broadcast, mobile signal, WiFi and GPS signals. For the DTV signals which I particularly studied, it ranges from 200 MHz to 900 MHz, and fits well within the unaliased range of the receiver.

The digitizers themselves have 8-bits precision, with an effective number of bits (ENOB) near 7.4. This produces a front end dynamic range of the order 40 dB. However, processing gain during down conversion performed in the FPGA substantially increases the dynamic range. For passive coherent radar the required instantaneous dynamic range can exceed 100 dB; this is achievable with the receiver we describe.

High speed samplers and digital signal processors offer greatly relaxed of requirements for analog filters. For the 5 GSPS samplers all signal power below 2.5 GHz is Nyquist sampled, and nearly any desired filter function can be realized in FPGA firmware. In our experience, each set of high performance analog filters costs about 20% of the cost of the rest of the data acquisition system, and would require manual exchange or additional RF switch fabric. By sampling sufficiently quickly the receiver complexity is greatly reduced and the flexibility retained.

For each RF input, the new receiver is capable of simultaneously extracting multiband channels from the fast sampler and FPGA processing ability. The first and second generation receivers were capable of extracting several FM stations from each antenna, but could not extract FM and DTV signals simultaneously without significant aliasing and undersampling challenges. With high speed sampling and rapid signal processing, VHF and UHF signals can be simultaneously extracted from the same samples.

Down conversion is performed numerically in the FPGA. Therefore, the two 90 degree phase-shifted oscillators can be designed to achieve nearly perfect quadrature in amplitude and phase over the entire frequency range.

The receiver is straightforward to operate once the FPGA programming has been performed. Our current FPGA code design also allows the tuning frequencies to be changed in a few milliseconds without interrupting data acquisition.

The initial development used the ROACH-1 running the ADC at 0.8 and 1.6 GSPS [22]. With the dual 2.5 GSPS sampler on ROACH-2, we have been able to sample signal at 2.5 GSPS (with up to four RF inputs) or 5 GSPS (with up to two RF inputs). Thus, higher frequency signals such as GPS and Wi-Fi waveforms can be used as illumination sources. Successful operation of the receiver requires care in the management of the resources of the ADCs, the FPGA, and the networking path out of the FPGA.

2.3 HARDWARE

2.3.1 *ROACH-2 Receiver Board*

Each ROACH-2 board contains a Xilinx Virtex-6 FPGA (Model: XC6VSX475T), and two ADC cards are attached to the board. Figure 2.1 shows the physical board with the lid opened. In our experiments with the fourth generation receiver we mainly used two dual 2.5 GSPS samplers for the ROACH-2 (with Xilinx Virtex 6 FPGA). These ADCs can also be operated as a pair of single input 5 GSPS samplers. The configuration of the receiver board is described in Appendix A.

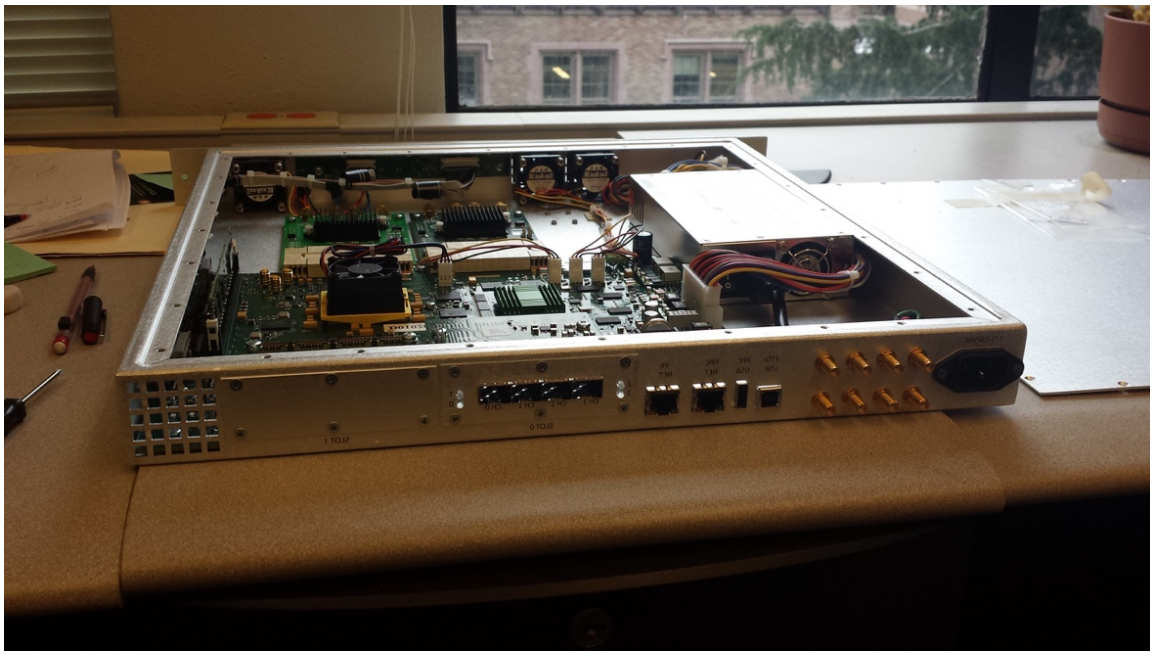
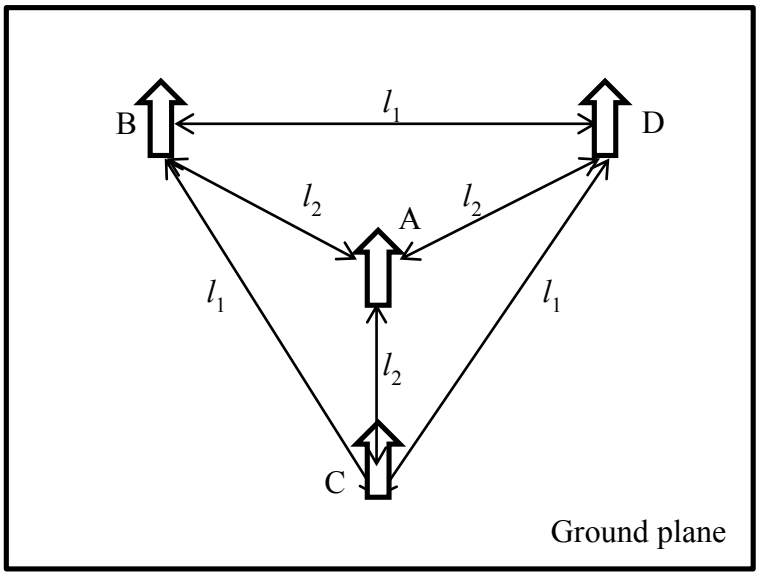


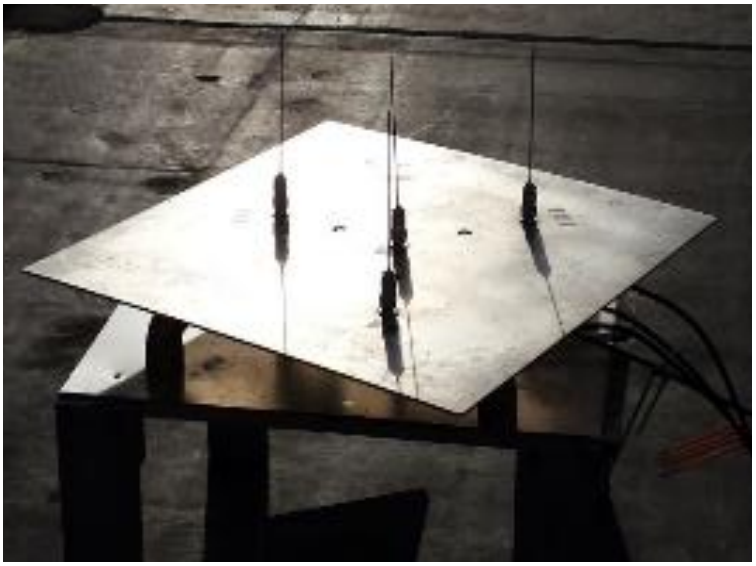
Figure 2.1. Physical board of ROACH-2 with lid removed.

2.3.2 *Antenna*

The antennas used are 4 simple monopole antennas, each 15 cm long. The four antennas are placed on a 61cm × 61cm metal ground plane. Figure 2.2 shows the block diagram and physical orientation of the antenna configuration. Three antennas (B-D) are placed at the vertices of an equilateral triangle with sides of length 31.5 cm (l_1). The fourth (A) is at the center of the triangle, which is 18.5 cm (l_2) from each vertex. Other configurations are possible; we selected this because it offers six independent baselines for interferometric analysis and imaging.



(a) Sketch of antenna array



(b) Photo of antenna array

Figure 2.2. Antenna configuration. The 4 antennas are monopole antennas, each 15 cm long. The four antennas are placed on a 61cm × 61cm metal ground plane. Three antennas (B-D) are placed at the vertices of an equilateral triangle with sides of length 31.5 cm (l_1). The fourth (A) is at the center of the triangle, which is 18.5 cm (l_2) from each vertex.

2.3.3 *Other RF Components*

Table 2.1 lists the properties of the analog RF components that are used in this radar system. The 2.5 GHz phase-locked oscillator is used to drive the FPGA and ADCs on ROACH-2 board. There are also the optional analog filters between the antenna and ADC, a 1.25 GHz low pass filter (LPF), or a UHF bandpass filter (BPF). A Low Noise Amplifier (LNA) and several fixed attenuators with fixed attenuation are also used to adjust the ADC input signal level to optimize the signal to truncation noise ratio.

The antennas are connected to a flexible 50ohm coaxial cable, optional low-pass filter, a series of 30dB Low Noise Amplifier (LNA), a series of fixed attenuator, ASIAA ADC (also called ADC5G [69]).

Table 2.1. Analog RF components

RF analog component	**Model**	**Company/Trademark**	**Properties**[1]
Phase-locked Oscillator (2.5 GHz)	SPC-SLFS-2500-02	EM Research®	Harmonics <20dBc; Spurs<-60dBc, Vcc=±5V, Phase noise<-100dBc @1kHz offset @1GHz
Low Pass Filter	5Mt-10-1500/1250-SM/SF	Microwave Filter Co. Inc.	f_{cutoff} = 1.25 GHz
Band Pass Filter	S/N 0702002	Microwave Filter Co. Inc.	$f_{passband}$ =500 MHz – 625 MHz
Amplifier (30 dB)	ZKL-2R5	Mini-Circuits®	Passband: 10MHz-2.5GHz; NF Typ. = 5dB; IP3 = 31 dBm (for output); Gain = 30dB; $VSWR_{in}$ = 1.4; DC power = 12V; 1dB Compr. Pout = 13dBm;
Attenuators	K2-VAT+ KIT	Mini-Circuits®	Fixed attenuation from 1dB to 10 dB; Passband: DC to 6 GHz; Maxmum input pwer: 1W
Coaxial Cable	3/8” Heliax	Andrew	Length = 25 meters
Coaxial Cable	RG 8		Length = 25 meters

[1] Note: Specification in properties are partially displayed here for UHF frequency range. More specification can be found in each component’s data sheet.

The amplifier (Mini-Circuits ZKL-2R5+) [70][1]is a low noise amplifier with 30 dB gain and 5 dB noise figure and has a wide passband from 10 MHz to 2500 MHz. The non-linearities are characterized by the 1dB compression point and the intermodulation distortion (IMD), especially the third order distortion (that is the third order interception point (IP3)). The 1dB compression point is the input power that causes the gain to decrease 1 dB from the normal expected linear gain plot, or the output power where the 1-dB drop occurs. It is the point where the amplifier goes into compression and becomes non-linear. For the 30dB amplifier, the 1dB compression point at the output is typically +13dBm (~1 Volts) for low frequency range [10MHz, 1250MHz] and +11dBm for high frequency range [1250MHz, 2500MHz]. Signal power received on the antenna need to be measured and maintained in order to make sure the amplifier operates below +11dBm for linear performance.

2.3.4 *ASIAA 5Gsps ADC*

Figure 2.3 shows the ADC daughter board. It contains two RF inputs, a sync port for synchronization with other ADCs and a clock port for the feed of 2.5GHz oscillator listed in Table 2.1.

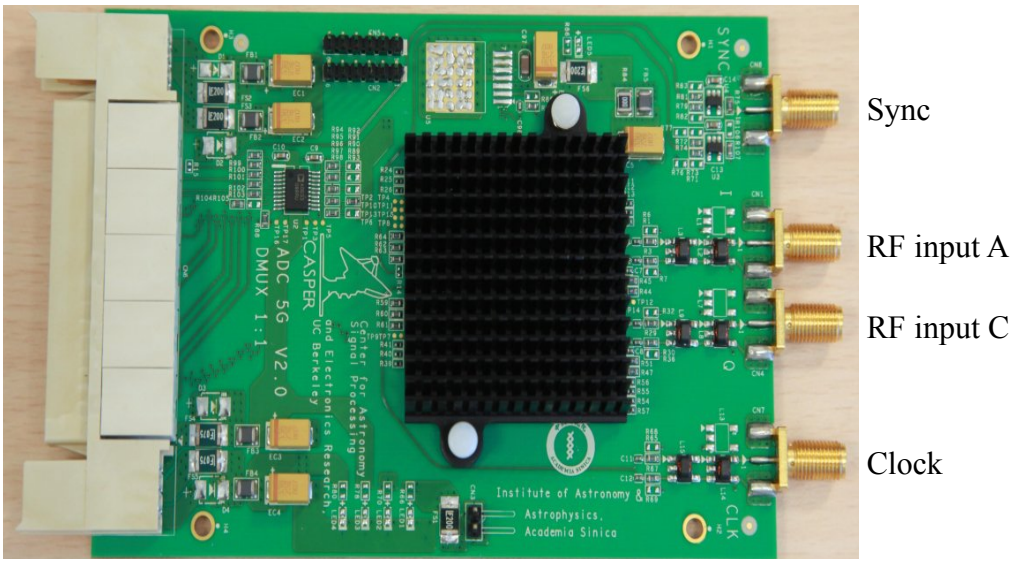


Figure 2.3. ADC daughter board. There are four input ports, one for synchronization; two RF input ports (A and C), one clock input. The while slot on the left is Mazzining card for connection with FPGA chip. There are two ADCs on the RAOCH board.

Each ADC board contains a single ADC chip (e2v^{TM} EV8AQ160 Quad ADC [69]) composed of four 8-bit ADC cores, each running at 1.25 GSPS. Figure 2.4 shows the block diagram of the chip. The board/chip can be run in two modes: two samplers for each running at 2.5 GSPS or a single sampler running at 5 GSPS. In 2.5 GSPS mode, two cores run interleaved and are assigned to the first sampler, while the other two cores run interleaved and are assigned to the second sampler. In the single 5 GSPS mode, four cores together run interleaved with 90-degree clock phase difference with its adjacent core.

Each ADC core has independent gain control (±18% range), offset control (±50 mV range) and phase control (±14 ps range); a calibration step is required to achieve the highest performance. It is possible to retrieve short bursts of the raw samples (1024 8-bit samples per RF input). Thus DC offsets and gain settings can be computed from the data; the sampler phases are adjusted by observing error counts ('glitches') from the sampler [69].

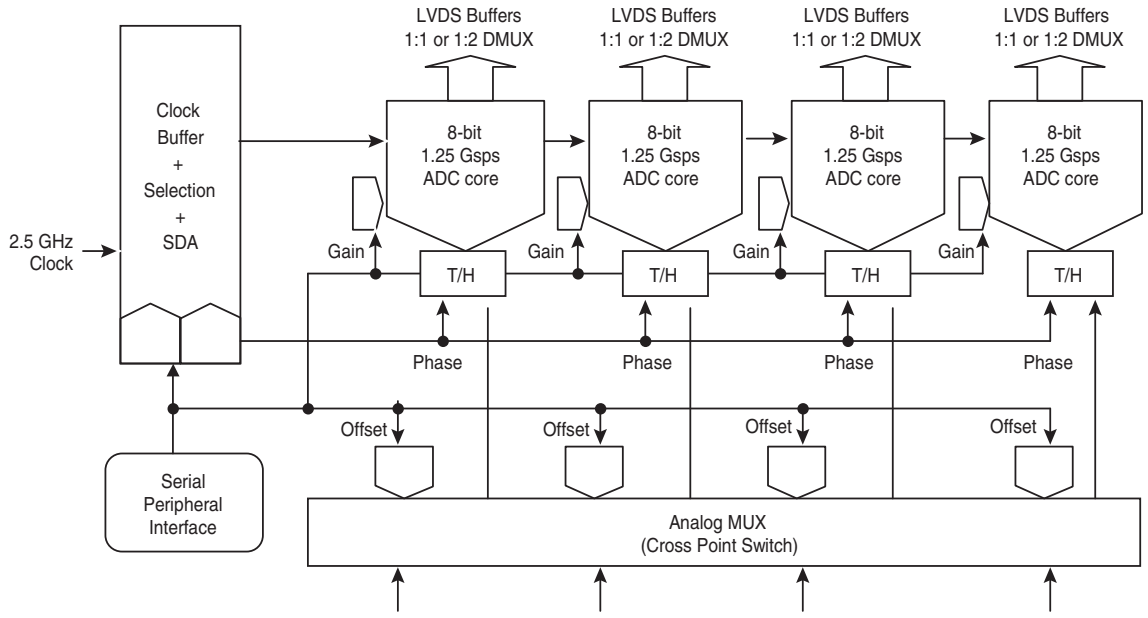


Figure 2.4. Simplified block diagram of the ADC [69]. There are four cores on the ADC, each clocked at 1.25 GHz. In 5 GSPS mode, four cores have 90 degree phase difference to produce 5 GSPS data stream. In 2.5 GSPS mode, every two cores are used to produce two 2.5GSPS data streams, and two cores for each stream have 180 phase difference. Each of the 4 cores has individual controls of gain, offset, and phase.

It is actually optimal for the input signal to use less than the full scale range when considering saturation and quantization noise. Considering Gaussian input, the RMS signal voltage should be the full-scale voltage divided by k, and for an 8 bit ADC k = 3.9, about 25% of full-scale [71-73]. This means we hope to be utilizing on the order of 65 voltage bins, about 6 bits (5 bits of amplitude). During the operation of the receiver, the gain is adjusted so that the RMS value from each sampler core is 35 (out of maximum of 128). This optimizes the tradeoff between quantization error and truncation error for Gaussian-distributed signals.

The full scale peak-to-peak in ADC data sheet is 500 mVpp, which indicates 250 mV maximum input voltage. The maximum input voltage level $V_{\max}$ is also tested with a signal generator, and $V_{\max}$ is measured to be 180 mV, slightly lower than the specification. It is probably due to the loss, impedance mismatch of the cable and adaptors. Therefore, the maximum received power in dBm:

$$P_{max} = 10 * log10\left(\frac{1}{2}\frac{0.25\text{volt}^2}{50\text{ohm}} \times 1000\text{ mW}\right) = 4.0\text{ dBm} \qquad (2.6)$$

For Gaussian signals, the average magnitude of input need to be kept at 1/3.9 of Vpp, or around -8 dBm input power. With appropriate attenuators, we can adjust the power that goes into ADC to be -8 dBm, which optimizes the dynamic range.

The ADC also introduces noise. For an ideal ADC with 8bit precision, its dynamic range is equal to the resolution, but in fact, the dynamic is reduced because of the truncation noise, non-linearity, aperture uncertainty, clock jitter, etc. The effective number of bits (ENOB), the number of bits that are on average not noise, is one measure used to characterize ADC dynamic range. The data sheet shows that the ENOB is typically 7.5 bits for 100 MHz single sine wave input at 2.5 GHz rate, and 7.2 bits for 620 MHz. The SNR[2] is typically 46dBc for f_{in}=100 MHz, and 44.5dBc for f_{in}=620 MHz.

Besides the thermal noise, nonlinearities cause harmonics, just like LNA. The total harmonic distortion (especially the first 25 harmonics) is quoted to be 58 dBc for 100 MHz input, and 55 dBc for 620 MHz input. The spurious-free dynamic range (SFDR), the ratio of the RMS (Root

[2] SNR: Ratio expressed in dB of the RMS signal amplitude, set to 1dB below full scale, to the RMS sum of all other spectral components including the nine first harmonics.

Mean Square) signal to the worst spur, is 58 dBc for 100 MHz input and 56 dBc for 620 MHz input with power of 1 dB less than full scale (1 dBFS). The two-tone third order intermodulation distortion is 50dBFS with f_1=490MHz and f_2=495 MHz (7 dBFS). The specifications for 5GHz mode and other frequency input can be found in the ADC data sheet [74].

2.4 DIGITAL SIGNAL PROCESSING

Besides the choice and setup of hardware, digital signal processing designs are extensively explored for the new receiver development, including DSP design in FPGA and DSP algorithm design for system performance improvements.

2.4.1 *DSP Design in FPGA*

After several alternatives were considered, the design for the FPGA adopts a digital implementation of a parallel I/Q mixer which digitally translates the desired sub-spectrum to baseband, followed by a polyphase FIR low-pass filter/downsampler, and finally a cascade-integrator-comb (CIC) filter to further reduce the bandwidth and data rate. Then the data pass out of the FPGA through one or more 10 GbE Ethernet ports. More details are discussed in Chapter 3.

2.4.2 *Post Data Processing*

Data from 10GbE ports are sent to Wenas, a computer server connected to ROACH-2 board. Wenas is a K10 NIVIDA GPU. On Wenas, data can be either real-time processed with the help of parallel computing, or stored to a disk to be processed later. For real-time processing, the real-time ambiguity was explored and revealed excellent performance with latency less than 1millisecond. For the non-real-time processing, various algorithms for the data were explored to characterize the system performance and to improve the detectability. More details about the algorithm design are discussed in Chapter 4 for dynamic range improvement.

2.5 SIGNAL TO NOISE RATIO (SNR) ANALYSIS

In this section, we present a study of the noise power from various noise sources in the analog components, ADC and FPGA.

2.5.1 *Front-end Noise Power and ADC Truncation Noise*

Let's compare the analog noise power at the ADC front end and the ADC truncation noise. To calculate the noise power of the front end, we first need to estimate the analog system temperature. Among the analog components preceding the ADC, we will assume that the LNA is the only component that generates noise. In other words, the lowpass filter is nearly ideal in its passband and no attenuator is used in the system. The assumption is approximately accurate based on the actual system setup. LNA has a noise figure of 5 dB (see Table 2.1), so the noise temperature T_e is about 627 K calculated from the following equation [21]:

$$NF = 10\log_{10}(1 + \frac{T_{\text{sys}}}{T_0}) \tag{2.7}$$

where T_0 is room temperature (290 K), NF is noise figure (dB) of LNA. Based on our assumption, the noise temperature T_e of LNA here is the system noise temperature T_{sys}.
Therefore, the noise power at the front end of the system is

$$P_{\text{n, sys}} = 10\log_{10}\left(kT_{\text{sys}}B\right) = -80 \text{ dBm} \tag{2.8}$$

Where k is Boltzmann's constant (1.38 x 10^{-23} J/K), B is 1250 MHz, the bandwidth of the 1.25 GHz analog lowpass filter.

On the other hand, with the knowledge of the maximum power of 4 dBm for single sine wave input of ADC and the SNR of ADC of 44.5 dBc, we can calculate the quantization noise of the ADC:

$$P_{quantization\ noise} = 4 \text{ dBm} - 44.5 \text{ dBc} = -40.5 \text{ dBm} \tag{2.9}$$

Compared to the system thermal noise, this shows that the largest noise source is the ADC quantization noise rather than the system noise. In other words, the effective temperature after the ADC is about 6,000,000 K. This suggests that a pre-amp with high NF but very high linearity would not impact the receiver performance. The number of bits or ENOB of the ADC in general defines the the system dynamic range. However, the dynamic range in our system is increased by the nature of the oversampling of the ADC, which we will discuss in section 2.5.2.

2.5.2 *Dynamic Range Increase Due to Oversampling*

The practice of sampling much faster than the frequency content of the signal of interest is called oversampling. This allows us to use an ADC with low resolution (fewer bits), and then gain more bits through signal processing [75]. A thorough explanation can also be found here [22].
In the case of our radar, we sample at 2.5 GSPS but we are only interesting in a 6 MHz wide DTV frequency band, oversampling by a factor of 400. This adds an effective 8.6 bits (in the ideal case), meaning we operate our 7.2 ENOB ADC as a 15.8-bit ADC (thus, the corresponding dynamic range is 15.8 x 6=94.8 dB). Due to the wideband lowpass filter and thus other frequency signals received at the same time, the signal of interest would not be full range of the ADC input, and therefore the SNR would be lower than 94.8 dB. Tracking of the SNR increase during the FPGA DSP processing is simulated and the results will be shown in Chapter 3.

2.5.3 *FPGA Truncation and Overflow Control*

Because of the fixed point operation of FPGA, the signal has finite number of bit during the processing stages in FPGA. Therefore, truncation and scaling needs to be carefully designed to avoid injecting additional truncation noise or clipping signals. In order to achieve as much processing gain as possible and allow a large dynamic input signal power, 16 bits is kept during the entire processing to avoid truncation/overflow issues introduced by essential fixed-point operation in the FPGA. In ROACH-2, a bit width of 16 bits is maintained during the processing chain. Simulations are used to validate that there is no additional truncation noise, and thus ensure sure that the only noise source comes from the ADC.

2.6 DTV Frequency Allocation and Aliasing Analysis

In this section, The DTV channels are explored and measured. Among those surrounding channels, we then selected two DTV channels that have relative large power and less frequency selective fading as the candidates of source of illumination for passive radar application.[3]

[3] For aliasing analysis, the bandpass filter in Table 2.1 is used in the system, instead of lowpass filter (1.25GHz).

2.6.1 *Frequency Analysis with FCC Reception Map*

First, we collected all the DTV stations according to the FCC reception map. On the map, we set the reception location at Sieg Hall where the receiver antenna is located. Table 2.2 lists received DTV channels with power larger than -17 dBm. Besides the received power of each channel, the TV towers location information for the first 6 strong channels, such as the distance from receiver, the azimuth angle relative to the receiver, the elevation and transmitting power, are also derived from FCC data and listed in Table 2.2. Figure 2.5 shows KONG DTV station reception map from FCC data [76].

Table 2.2. Surrounding DTV channels[4] with received power larger than -17 dBm

Channel name	Received power (dBm)	Center frequency (MHz)	Distance from Rx (meters)	Azimuth angle from Rx (degree)	Tx tower Elevation above sea level (meters)
KONG	-12	575	4200	215	258
KZJO	-9	539	4300	164	326
KOMO	-11	617	4300	216	300
KIRO	-11	623	4500	220	297
KSTW	-11	201	4300	165	311
KING	-16	677	4200	215	147

[4] The data is predicted by using FCC Mapping Tool (https://www.fcc.gov/media/engineering/dtvmaps). "These predictions are based on a terrain-sensitive propagation model resembling but not identical to the propagation model used when calculating service and interference contours for licensed broadcast television stations. Actual signal strength may vary based on a variety of factors, including, but not limited to, building construction, neighboring buildings and trees, weather, and specific reception hardware. Your signal strength may be significantly lower in extremely hilly areas."

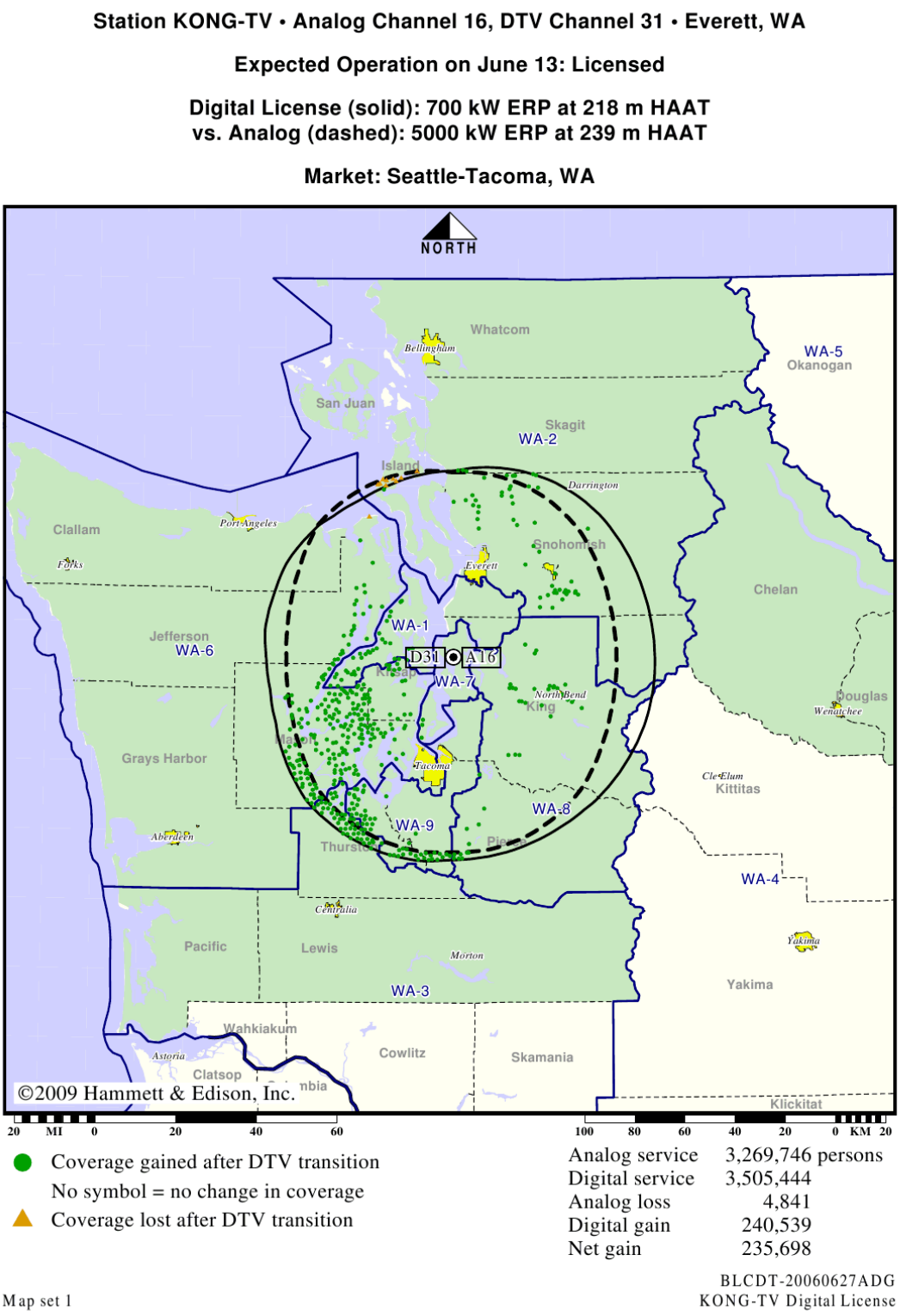


Figure 2.5. DTV reception map from FCC data [76].

2.6.2 *Frequency Distribution Measured with Spectrum Analyzer*

Then we measured the spectrum of signals that our receiver monopole antenna can capture. Figure 2.6 below shows the spectrum across DC to 3 GHz measured with Keysight™ FieldFox (model N9914A [77]) in its Spectrum Analyzer mode.

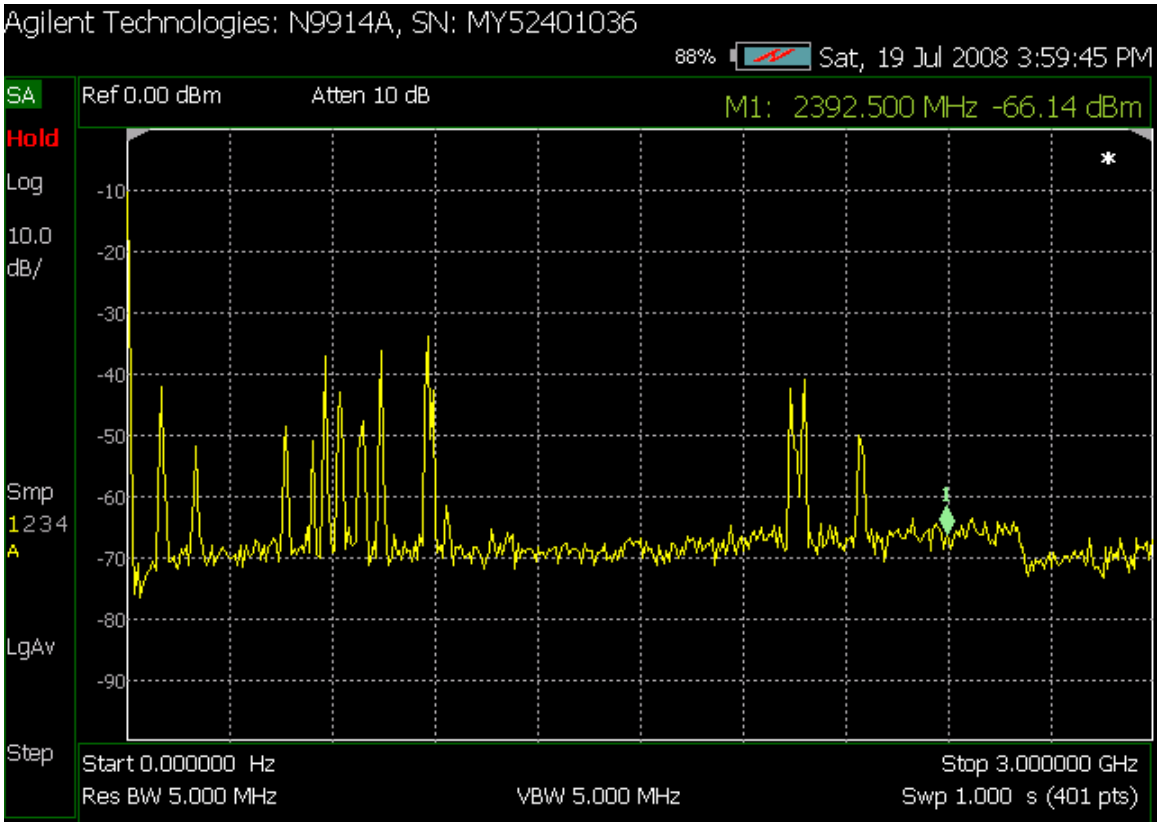


Figure 2.6. Frequency spectrum across DC to 3 GHz measured with FieldFox Spectrum Analyzer. The frequency sweeps from 0 – 3 GHz.

The DTV portion of the spectrum with the range of 400MHz-950MHz is expanded in and shown in Figure 2.7. The second one is identical to the first one except adding some markers. The signals between 700MHz-950MHz are allocated for mobile communication, so here we focus on the signal between 400MHz and 700MHz that our monopole antenna is optimized to.

The available DTV signal and the power of these DTV signals are slightly different from the FCC data because of the actually exposed environment and use of different antennas. There are two mobile channels around 460MHz, KZJO DTV channel at 539MHz, KONG DTV channel at

575MHz, KOMO and KIRO DTV channel around 620MHz, two channels around 683MHz (broadcast or mobile). Because an analog BP filter of 450-650MHz is adopted in the receiver for avoiding signals aliased to the KONG channel, we currently focus on KZJO, KONG, KOMO and KIRO channels. Table 2.3 lists the transmitter information. However, in the future we may consider removing the filter to be able to use additional channels.

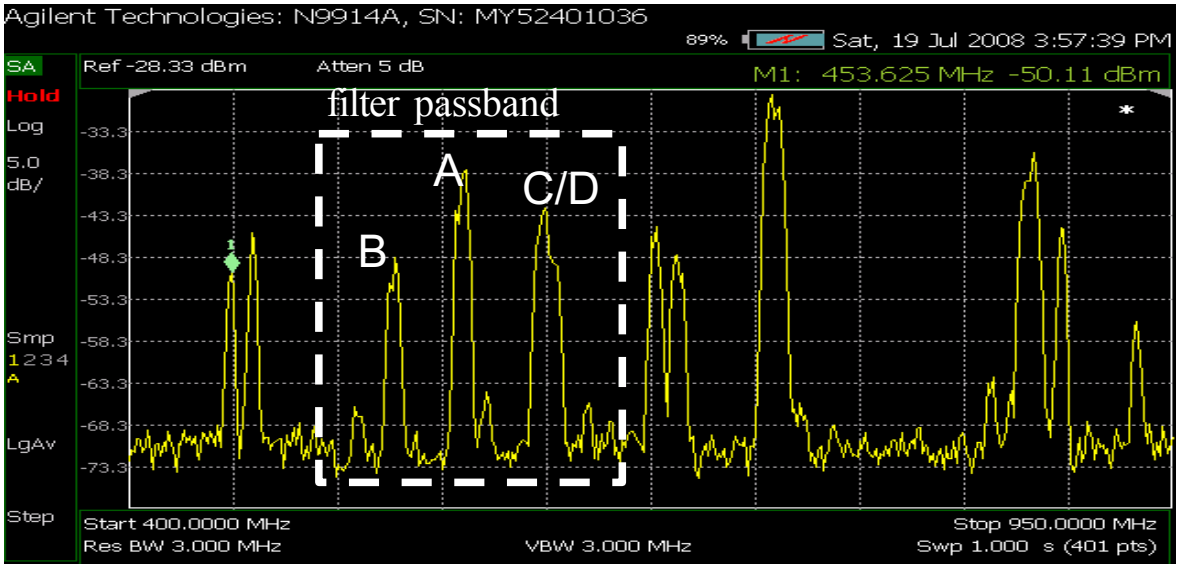


Figure 2.7. Expected DTV channels within the range of 400MHz-950MHz measured with FieldFox spectrum analyzer. There are quite a few DTV channels in this range. They also have relatively high SNR.

Table 2.3. DTV channel and locations

Label in Figure 2.7	DTV Channel	Center Frequency (MHz)	Distance from Rx	Azimuth angle (respect to Rx)	Tx Tower Elevation (above sea level)	ERP
A	KONG	575 MHz	4,200 m	215 º	258 m	700 kW
B	KZJO	539 MHz	4,300 m	164º	326 m	1000 kW
C	KOMO	617 MHz	4,300 m	216º	300 m	1000 kW
D	KIRO	623 MHz	4,500 m	220º	297 m	925 kW

2.6.3 *Receiver Leakage and Aliasing Analysis*

Leakage and aliasing is analyzed from the raw data sampled by the 2.5GSPS ADC. We used the bandpass filter (500 - 625 MHz) in the test, which is true for using DTV signal in our later work in Chapter 3 and 4. The raw data is captured in FPGA. After the signal is digitized with the 2.5GSPS ADC, the raw samples are sent to the FPGA. The FPGA captures a small amount of data, usually 8192 points, and saves it to the internal RAM for the host computer to retrieve.

Figure 2.8 shows the spectrum of the raw data. We still can see some some FM broadcast and high frequency signals above 1 GHz, which indicates that these signals were leaked through the Bandpass filter. The low frequency signals among the leakage, such as FM at around 100 MHz and signals at 200 MHz, would still shows at their original frequencies, as indicated in Figure 2.8. Those leakages at high frequencies, such as 2.14 GHz and 1.96 GHz, would fold into 360 MHz and 540MHz, as listed in **Error! Reference source not found.** We can see the signal at 360 MHz in Figure 2.8, which does not exist in the FieldFox SA result. The other aliased signal at 540 MHz is overlapped on to the KZJO DTV channel at 539 MHz.

Table 2.4. Analog bandpass filter leakage

Leaked Frequency Channel	Bandwidth	Possible Source	frequency after alias (fs: 2.5GHz)	Affected DTV stations
2.14 GHz	30 MHz	Mobile	360 MHz	N/A
1.96 GHz	60 MHz	Mobile	540 MHz	KZJO @539MHz
FM broadcasts	20 MHz	FM broadcast	No alias	-
200 MHz	6 MHz	DTV broadcast	No alias	-

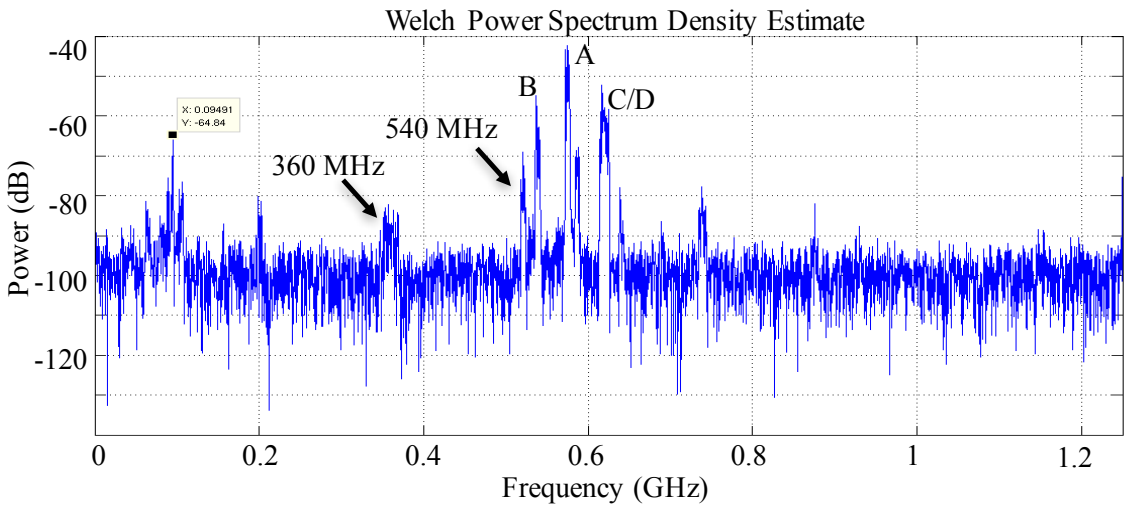


Figure 2.8. Spectrum of the raw data sampled by the 2.5 GSPS ADC. The (500 MHz – 625 MHz) bandpass filter are used in the test, but some signals are still observed in the stop band.

We expanded the frequencies (1.9 GHz to 2.2 GHz) of the Fieldfox spectrum measurement as shown in Figure 2.9, to take a close look at the two high frequency signals. The one at 2.14 GHz has a bandwidth of 30 MHz; the other at 1.96 GHz has bandwidth of 60 MHz. The 1.96 GHz shares the high band cell phone activity.

After the 2.5GHz sampler, these two signals are aliased to 360 MHz (BW= 30 MHz) and 540 MHz (BW= 60 MHz). The one at 360 MHz is the signal that have been seen in the raw data spectrum Figure 2.8. The signal at 540 MHz (range of 510-570MHz) is less noticeable in the raw data spectrum, but to be safe, it would be better to not use KZJO channel at 539 MHz. In the future, investigation can be made by taking data without filter and see how strong the original signal is.

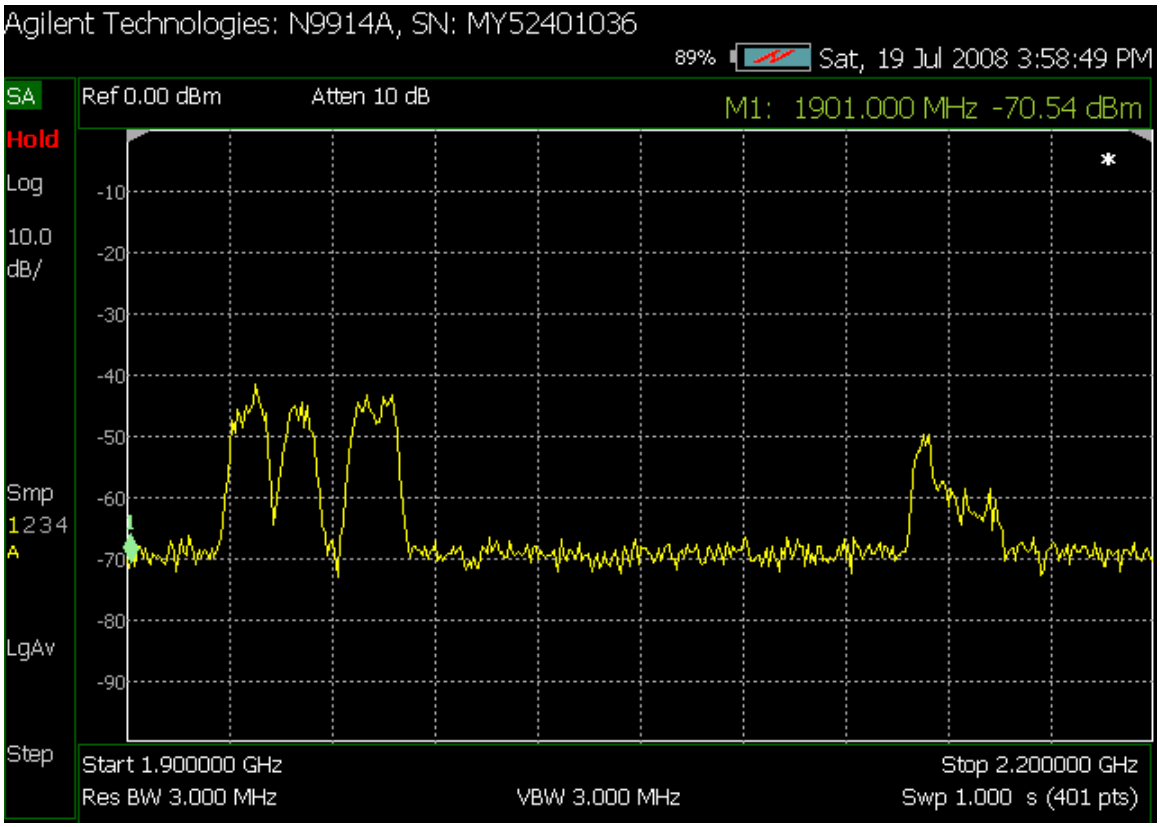


Figure 2.9. Spectrum across 1.9GHz to 2.2 GHz measured with Fieldfox spectrum analyzer. The signals at 1.96 GHz with 60 MHz bandwidth and the signal at 2.14 GHz with 30 MHz are observed. Those two signals are aliased to frequency range of interest after digitizer.

2.7 SUMMARY

In this chapter, we gave an overview of the system architecture and introduced the hardware of our receiver. We analyzed the signal to noise ratio with noise from the environment and the digitizer, and showed that the digitizer noise is much higher than the environmental noise even of full bandwidth. We analyzed the signal presented at our antenna, especially the strength of several DTV signals. We showed that due to insufficient stop band suppression of the bandpass filters, unwanted signals are folded onto some DTV signals. Considering the strength of DTV signals and aliasing effects, we showed that the KONG DTV signals at central frequency of 575MHz is a good candidate for illumination in the sense of passive radar.

Chapter 3. ROACH-2 (RECEIVER) DESIGN

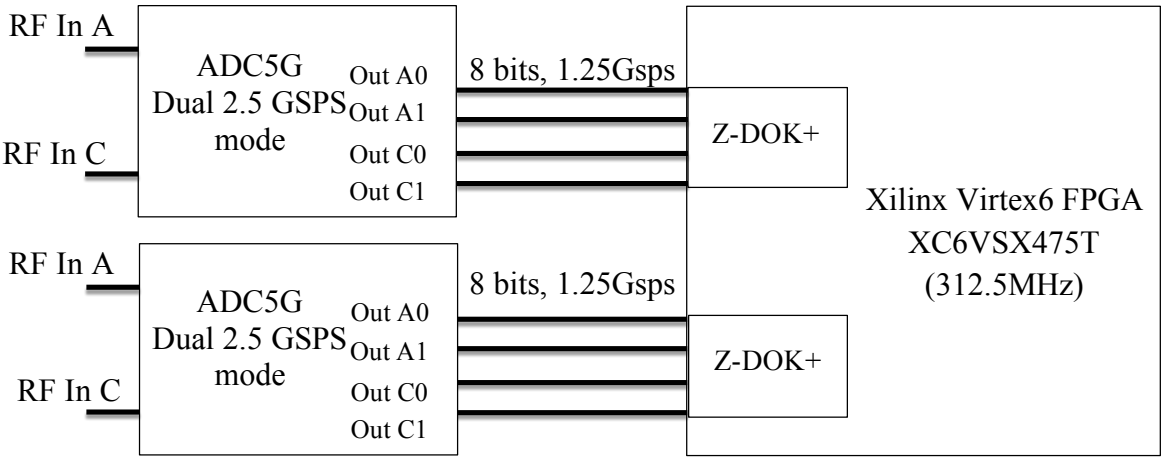
In this chapter we will discuss the FPGA design of our new receiver. The signal flow is first introduced and then discussed in detail. Arithmetic operation and design limitations are discussed. We simulate the design in Simulink and the SNR is examined to verify that bit width and truncation is selected appropriately. Finally, several practical designs are introduced, and the results are demonstrated and compared.

3.1 SIGNAL FLOW CHAIN

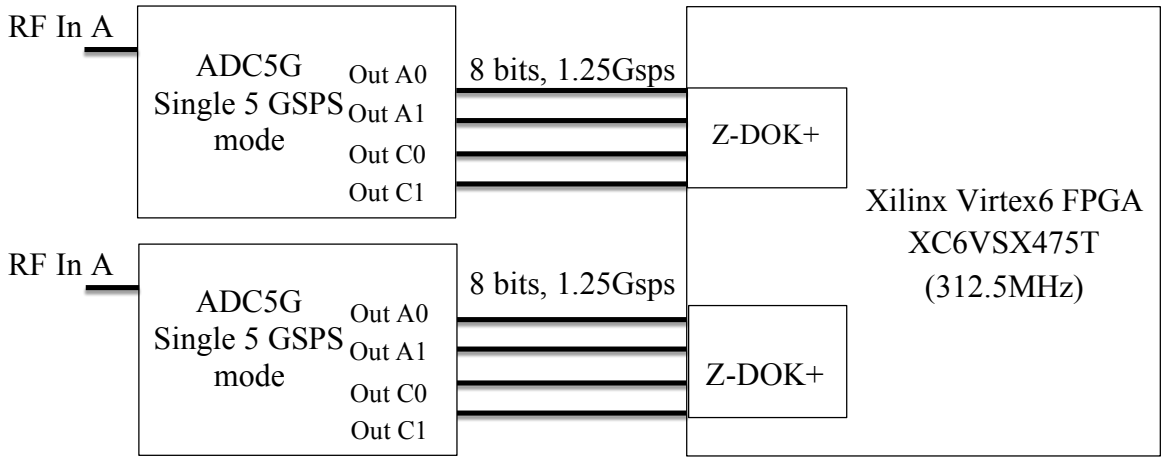
Each ROACH-2 board contains a Xilinx FPGA, and up to two ADC cards may be attached to the board via ZDOK+ connectors. In our experiments with the fourth generation receiver we used two dual 2.5 GSPS samplers for the ROACH-2 (with Xilinx Virtex 6 FPGA). These ADCs can also be operated as a pair of single input 5 GSPS samplers. A block diagram of the receivers is shown in Figure 3.1. In this work, we emphasize the dual 2.5 GSPS mode.

The front end configuration of the digitizer varies; in practice, in the urban setting at the University of Washington, we have found that the signal voltage provided by the antenna is sufficient to drive the digitizers directly; it is of course possible to insert amplifiers or attenuators as needed.

After several alternatives were considered, we adopted a design for the FPGA that is a digital implementation of a (parallel) I/Q mixer which digitally translates the desired sub-spectrum to baseband, followed by a polyphase FIR low-pass filter/downsampler and finally a cascade-integrator-comb (CIC) filter to further reduce the bandwidth and data rate. Then the data pass out of the FPGA through one or more 10 GbE Ethernet ports; we used a single 10 GbE port for this study. Figure 3.2 shows the processing flow schematically.



(a) Dual 2.5GSPS mode



(b) Single 5GSPS mode

Figure 3.1. Block diagram of the fourth generation MRR receiver. (a) Dual 2.5GSPS mode of the ADC. (b) Single 5 GSPS mode of the ADC. The signal path from ADC to FPGA uses low voltage differential signaling (LVDS). For Dual 2.5GSPS mode, signals collected by RF In A is split into two discrete data streams and sent through Out A0 and Out A1. For Single 5GSPS mode, signal from RF In A is sent through Out A0, Out A1, Out A2 and Out A3.

Figure 3.3 illustrates spectrum changes in each stage. Because the very first step is I/Q full rate down conversion, complications due to aliasing are minimized. Assume there are two frequency signals at f_0 and f_1 (as shown in Stage 1 of Figure 3.3) received by the antenna, where f_0 is the signal that we are interested in, and f_1 is the signal we want to remove from the data. After the first stage of mixing operation the positive f_0 spectrum is shifted down to baseband. In the second stage a low-pass polyphase filter suppresses the negative f_0 and other unwanted spectral content, leaving only the positive f_0 spectrum in in-phase-and-quadrature form. In the CIC filtering stage, the data rate is further reduced according to the bandwidth of the f_0 signal and then exported from the FPGA through high-speed 10 GbE interfaces.

Figure 3.2 and Figure 3.3 show the processing chain for one antenna and one selected RF channel. In practice this is replicated for each antenna and RF channel combination. The ROACH-2/Virtex 6 is capable of handling three RF channels, each with bandwidth 150 MHz on each of four antennas with our implementation.

Figure 3.4 illustrates the data propagation inside the FPGA. First, x_t is the digitized 8-bit sample at 2.5 GHz; x_t is then divided into 8 parallel streams with data rate at 312.5 MHz and fed into a DDS (Direct Digital Synthesizer) block. The DDS generator produces discrete sine and cosine waveforms ys_t and yc_t with frequency f_0. In the DDS block, the data x_t is multiplied with ys_t and yc_t respectively, producing in phase and quadrature (I/Q) data whose frequency is moved to baseband. zi_t and zq_t represent the I and Q samples. Then the I/Q data pass through a polyphase low-pass filter to reduce the data bandwidth to the FPGA clock speed for downsampling. Next, a 6th order CIC filter reduces the data rate by an additional adjustable factor R. Finally the I/Q data (si_t, sq_t) are scaled, truncated, combined, and then sent to the 10 GbE buffer to form UDP packets which are sent out for subsequent processing.

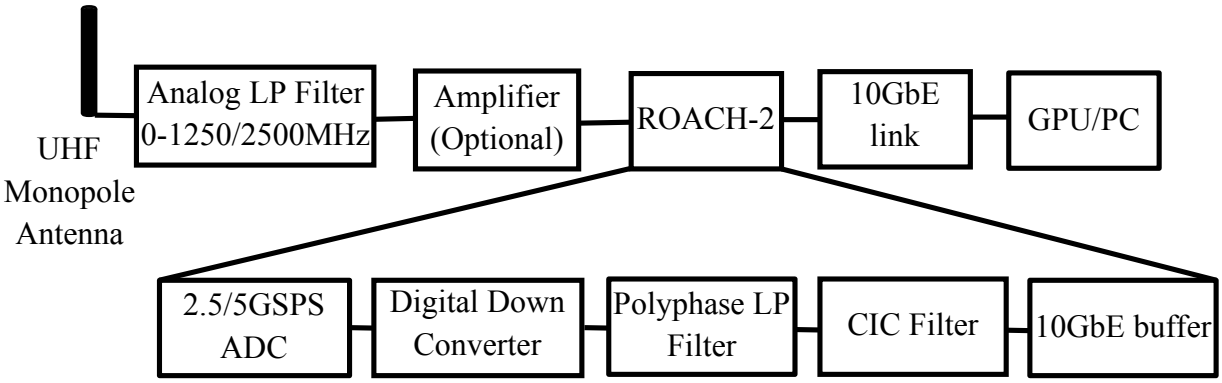


Figure 3.2. The Block Diagram of the receiver architecture. The signal is first received by a monopole antenna, and passes through analog lowpass filter and amplifiers. In ROACH-2 board, the signal is digitized by the 2.5/5GSPS ADC, is digital down converted to base band, filtered and decimated with polyphaser lowpass filter and CIC filter, and finally sent out through the 10Gb Ethernet network. Data is saved in GPU/PC which can also perform further data analysis offline or in real time[48].

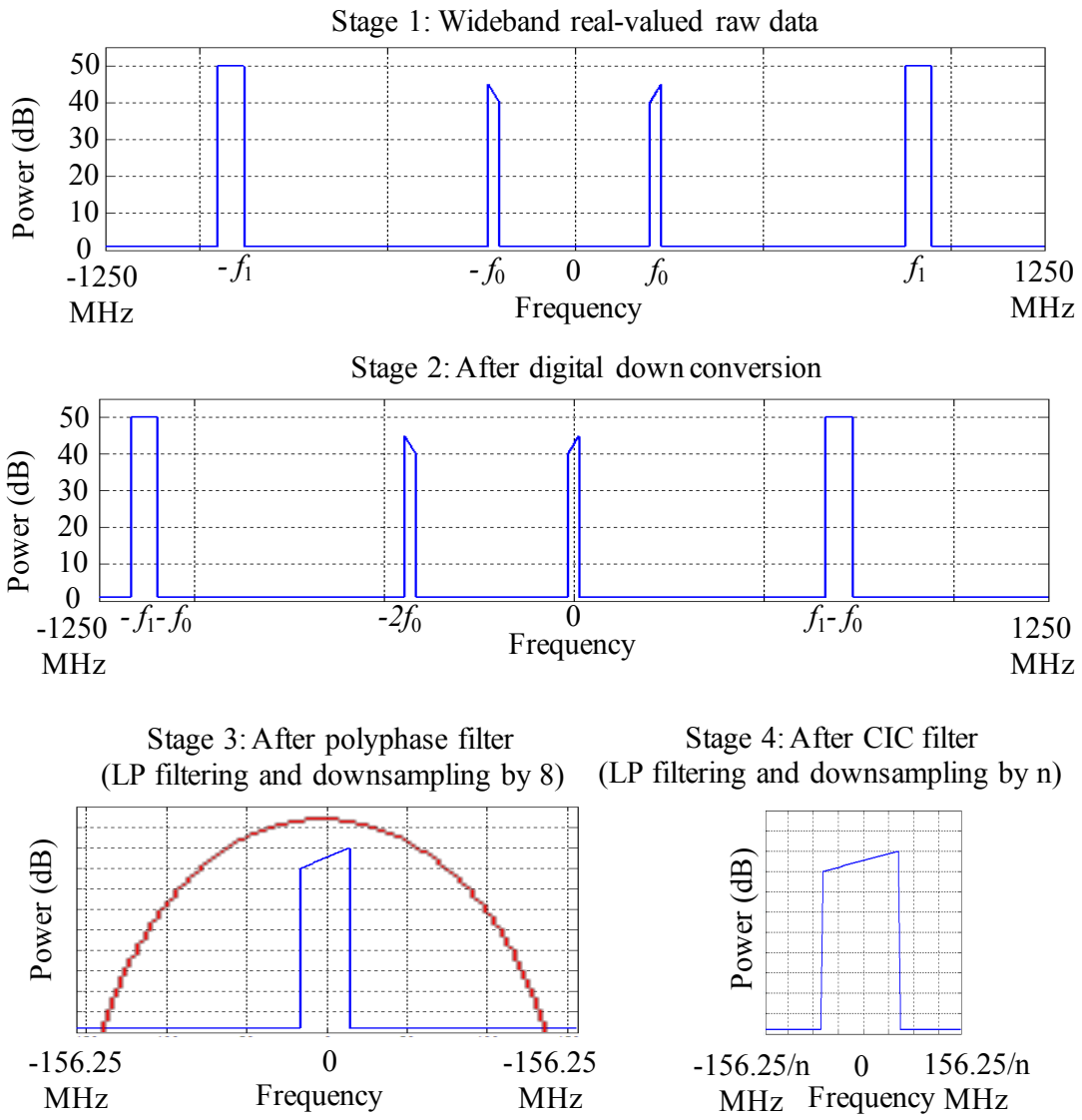


Figure 3.3. Spectrum representation of the processing flow. It involves four stages. Stage 1 shows the raw spectrum of data digitized from ADC. The signal at f_0 is the one of interest. In order to show the filtering effect, an unwanted signal at f_1 is added. State 2 shows the spectrum after signals are down converted by f_0. The signal at f_0 is shifted to baseband (DC). The signal at $-f_0$ and signal at f1 are also shifted to the left by f_0. Stage 3 shows the spectrum after signals are filtered and decimated by 8 with the polyphase lowpass filter. The red curve represents the filter frequency response. The unwanted signal gets attenuated and removed in this stage. Stage 4 shows the spectrum after the signal is further decimated by n with the CIC filter. Signals near f_0 which are not demonstrated here would filtered out and removed by the CIC filter.

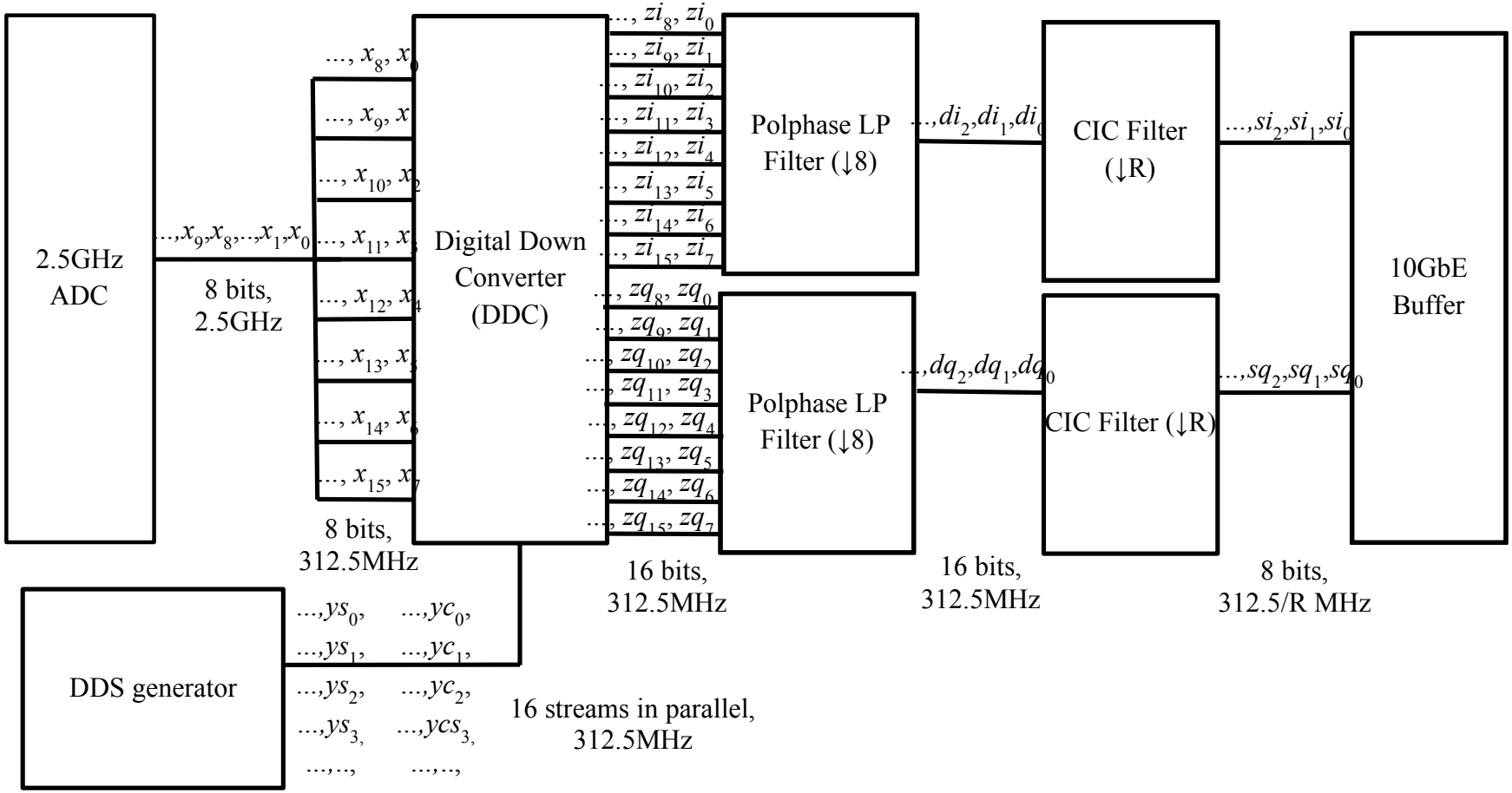


Figure 3.4. Diagram of data propagation inside the FPGA. *xi* represents the digitized raw samples. It goes into the FPGA in 8 parallel streams (or 16 parallel streams if the ADC 5GSPS mode is used). After DDC the signal becomes baseband Inphase (*zi*) and Quadrature (*zq*). Two identical Polyphase lowpass filters are applied to the I/Q data and output di and dq which are further filtered and decimated by two identical CIC filters. Finally the decimated signal *si* and *sq* is sent to 10GeE buffer for transmission.

3.2 MANAGEMENT OF ADC

3.2.1 *ADC Yellow Block*

The Analog to Digital Converter (ADC) of our system is DMUX 1:1 version of 5Gsps ADC which comes with the ROACH2 board. The DMUX 1:1 version converts analog signals to 8-bit samples. The clock required is 2.5GHz. It works at 8-bit Single 5GSPS ADC in one channel mode, or 8-bit dual 2.5GSPS in two channel mode. The current design mainly discussed here is in two channel mode. It takes in two streams of continues analog signal, i.e. two RF signals at 2.5GHz, and convert each RF signal into 8-bit samples which will be automatically split into 8 parallel streams for each RF signal in the FPGA, as shown in Figure 3.4. Figure 3.5 shows the ADC yellow block in Matlab Simulink environment. As mentioned in Chapter1, the yellow blocks developed by CASPER community include the configurations of physical interface. For ADC yellow block, it receives digitized signals from ADC and outputs them to FPGA fabrics which are represented as ports from a0 to a7 and b0 to b7 in the yellow blocks. The inputs 'sim_a' and 'sim_c' are only used for simulation. The differential signals from ADC is unsigned 8 bit value, and in the white block followed by ADC yellow block, the signal was converted to 2's complement value (eight parallel samples are cncatenated into a single 64 bit value as shown UFix_64_0 in Figure 3.5. Then the signal is captured in the Snapshot RAM which is used to examine the wideband raw signal from ADC.

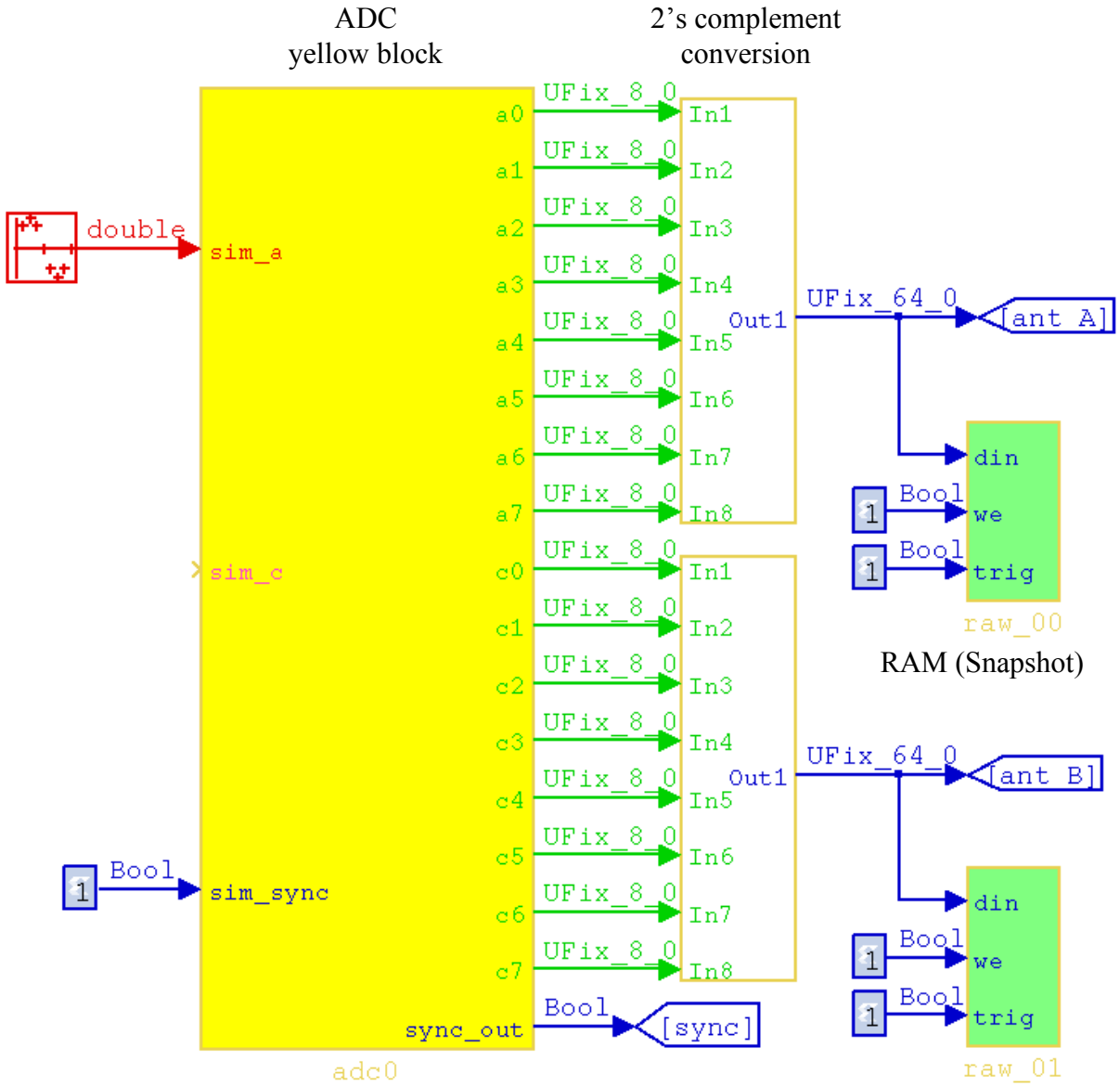


Figure 3.5. ADC yellow block. The input port 'sim_a' and 'sim_c' of the yellow block take in simulated signals in Simulink, and the actual input ports would take in signal from SMA connectors. The ports a0-a7 and c0-c7 output 8 parallel streams from each of the two input in the 2.5 GSPS mode. In 5 GSPS mode, the total 16 parallel streams are the digitized signal from a single input. Followed with ADC is the 2's complement conversion which converts the unsigned 8-bit sample into 2's complement form. A snapshot block (green) is inserted after 2's complement conversion for each signal channel to store a small amount of raw samples for analysis. Note that the signal is represented in fixed point. For example, 'UFix_8_0' represents unsigned fixed-point with 8-bit integer and 0-bit fraction.

Figure 3.6 shows the parameter configuration of ADC yellow block. It offers users the choice of ADC modes (5 or 2.5 GSPS), sampling rate, etc. The actual ADC clock operates at 2500 MHz, and the corresponding FPGA clock is at 312.5 MHz. However, the software doesn't allow selecting this value for simulation. The development platform only offers limited available clock values, and 2500 MHz is not among these values. In order to work around it, we selected the avalaible 2704 MHz, a value that is closest to 2500 MHz value and higher than 2500 MHz for design. When the system actually operates at 2500 MHz. the design has less chance to cause problems than a design below 2500 MHz. Therefore, the sampling rate is set to 2704 MHz, and the FPGA clock which is 1/8 of ADC clock is thus set to 338 MHz. The system operates with a 2.5GHz clock oscillator, i.e. 2.5GHz for the ADC and 312.5 MHz for the FPGA.

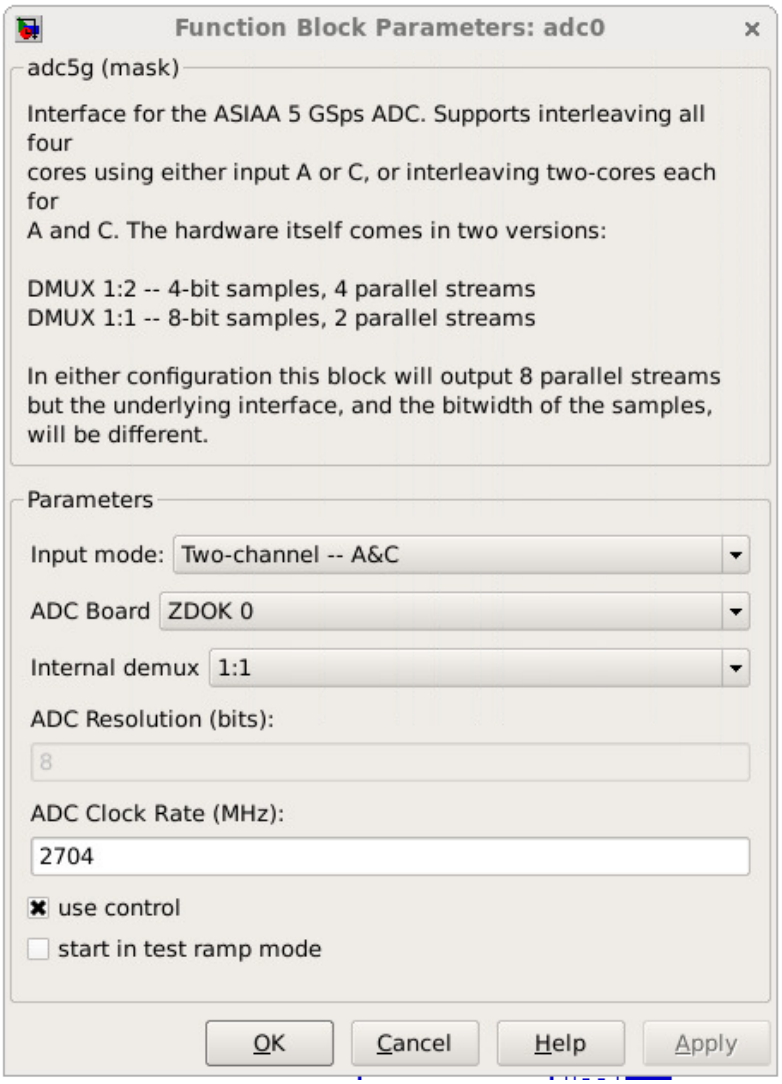


Figure 3.6. ADC yellow block parameter configuration. Some important parameters are input mode (2.5GSPS or 5GSPS), physical ADC board that the yellow block represents and the ADC clock rate. The ADC Clock Rate is set to 2704 MHz for simulation and compilation, while the ADC is actually running at 2500 MHz.

The 8-bit sample from the sampler is interpreted in the form of unsigned number, known as offset binary. The unsigned number has to be converted into two's complement signed number. The correction is achieved by inverting the MSB (the most significant bit) (shown in Figure 3.7). Therefore, the corrected sample is an integer between -128 to 127.

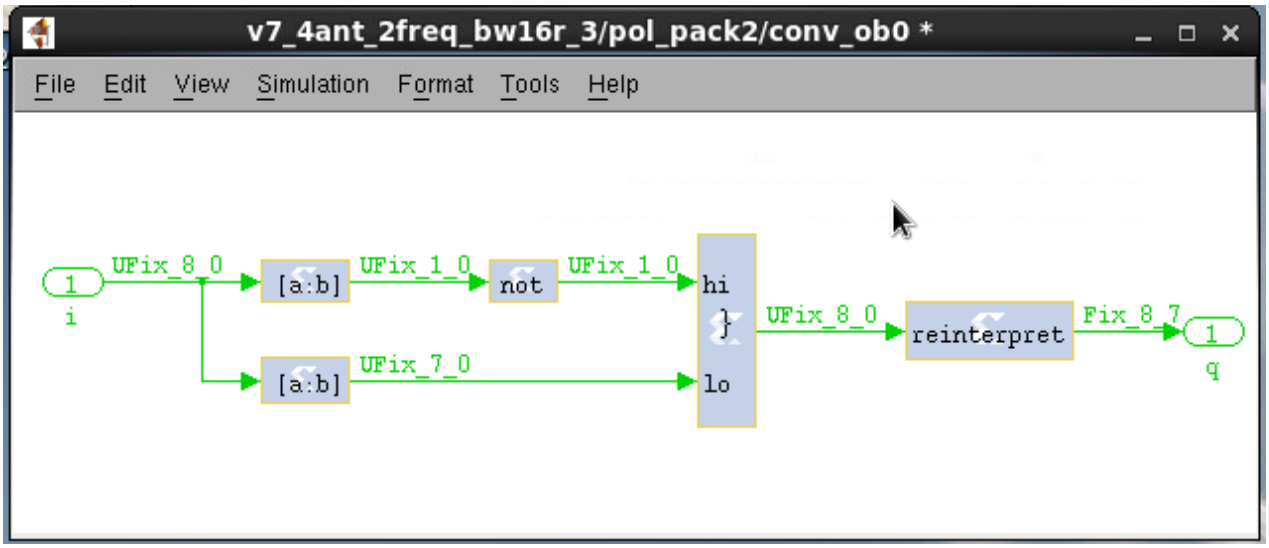


Figure 3.7. The block of two's complement conversion for the digitized raw data. The fundamental of this block is to flip the most significant bit (MSB). In the top path, the slicing block '[a:b]' slice the MSB and flip the sign with 'not' block. the bottom slicing block slice the rest 7 bits. The two signals is concatenated into 8 bits in the concatenation block and then reinterpreted (scaled by 2^7) from 8-bit integer to the value with 7 fractional bits.

3.2.2 *Truncation and Spurious Free Dynamic Range*

A snapshot block after the data type correction is used to grab a small amount of full rate samples during run time. A test is made to measure the signal to truncation noise ratio. Figure 3.8 shows the spectrum of the snapshot samples with Welch Method. The ADC is fed with a 101MHz sine wave from a signal generator. The magnitude of the digitized sin wave is measured at 100 out of 128. From the figure, we can see that the average noise floor is about 80 dB below the signal. There are 8192 samples stored in the snapshot, so the processing gain of FFT is 36 dB calculated with Eq. 3.10 where M is the number of FFT point:

$$\text{FFT processing gain} = 10\log_{10}\left(\frac{M}{2}\right) \tag{3.10}$$

Therefore, the theoretical SNR is 80 dB – 36 dB= 44 dB, which agrees with the data sheet.

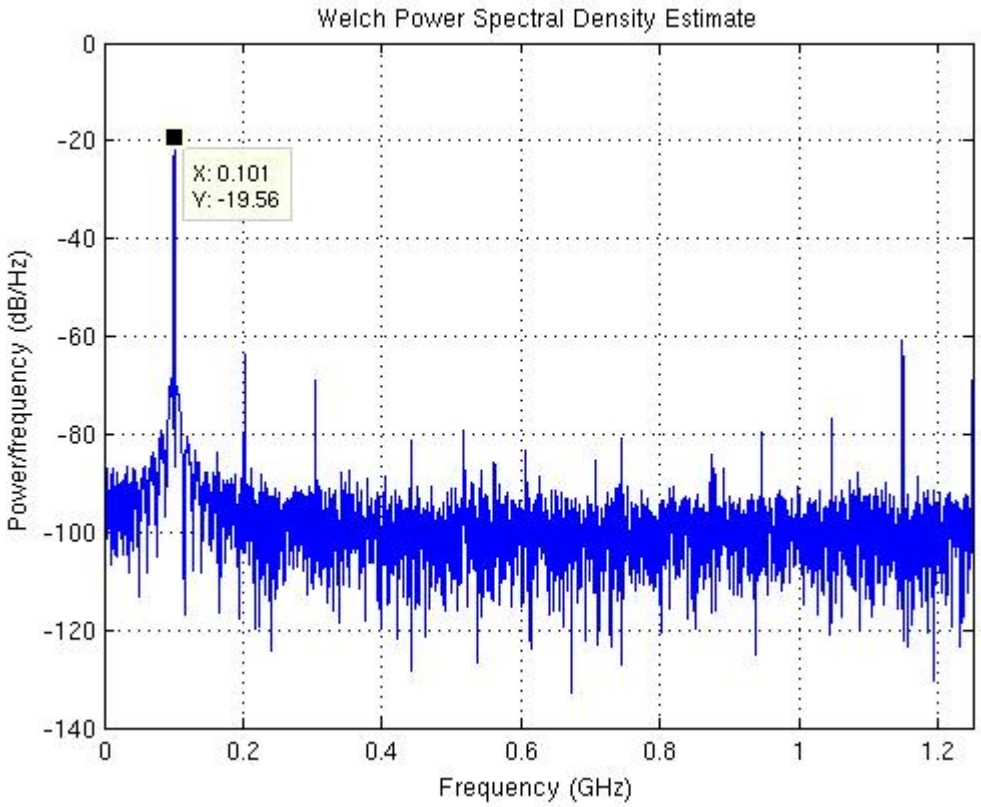


Figure 3.8. A snapshot of the raw data from ADC fed with a 101 MHz sinusoidal signal. The Nyquist frequency is 1.25 GHz as the ADC operates in 2.5 GSPS mode. Besides the signal peak at 101 MHz, we can also see the 2nd and 3rd order harmonics at 202MHz and 303MHz, which are mainly due to the ADC's nonlinearity.

3.2.3 *ADC Calibration*

A. Offset Calibration

Assume the noise present at the ADC inputs is zero mean. The offset was measured with input ports open circuited. This test is for dual 2.5 GHz mode. In this mode, two of the four ADC cores sample one input interleaved. One stream signal from single input is split into 8 parallel streams at the output.

The procedure of the test is as follows: first disconnect the cable from ADC, and take some raw data with snapshot; then measure the noise level and also compare the noise property of the two cores which clock interleaved to produce the 2500MHz samples.

Figure 3.9 shows the raw samples presented in the 8 parallel streams. The horizontal axis is in unit of the sample in the order of time, and the vertical axis is the sample amplitude. From the top left to the bottom right are the 8 streams in order. Figure 3.10 shows the histogram of the two cores. The samples in the even number of the streams are from core 0, and the samples in the odd number of the streams are from core 1. It's clear that the offset of the two cores has been well adjusted, since most values are 0. However, the noise distribution of the two cores are slightly different.

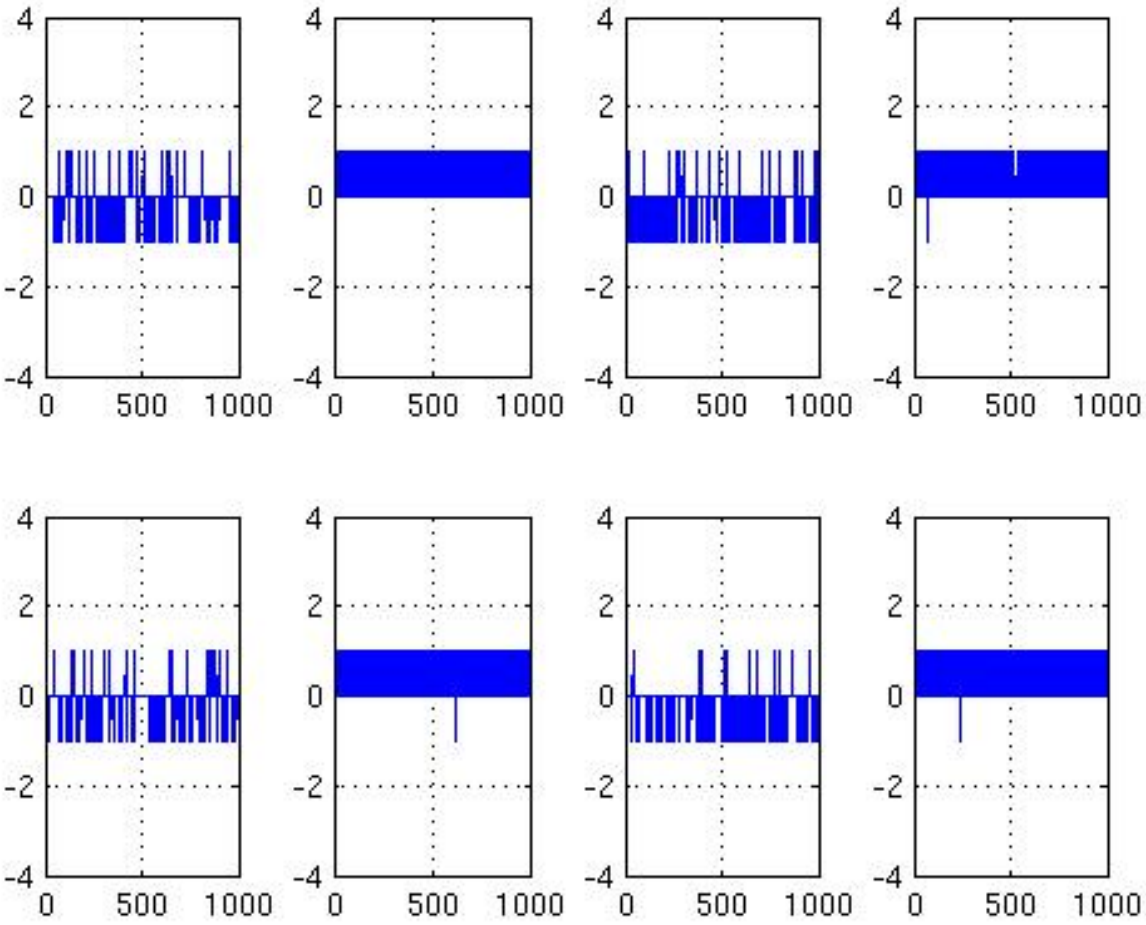


Figure 3.9. The raw samples presented in the 8 parallel streams after the offset is well adjusted. horizontal axis is in time, and vertical axis is the amplitude. From the top left to the bottom right are the 8 streams in order. Data from the odd number figures are generated from core 1, and data from the even number figures are generated from core 2. Both core has an offset of 0, but with different statistical distribution.

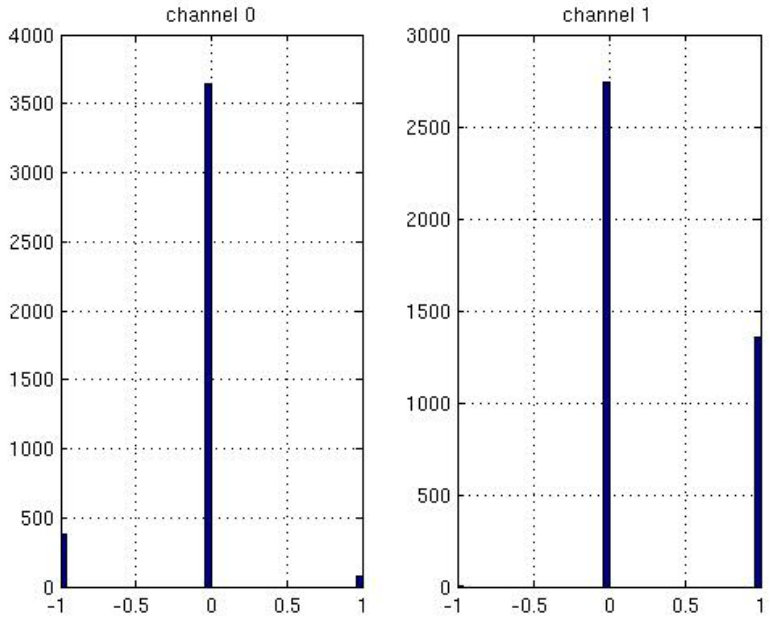
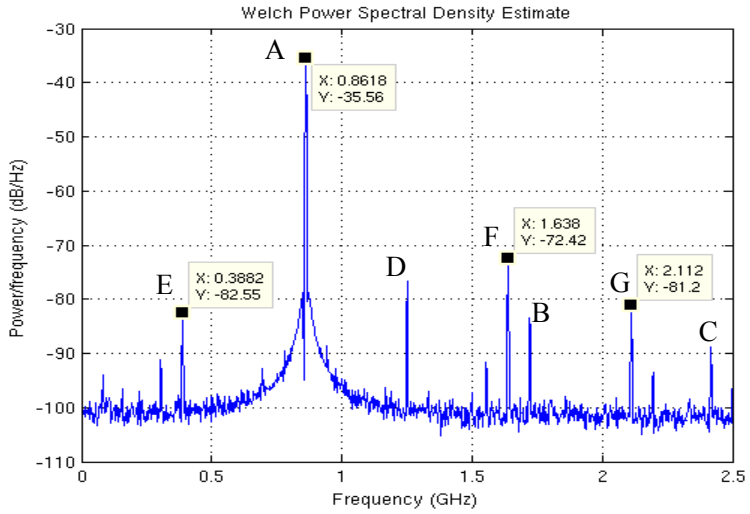


Figure 3.10. The sample histogram of the two cores. The sample distribution also indicates a well-adjusted offset, and a different distribution.

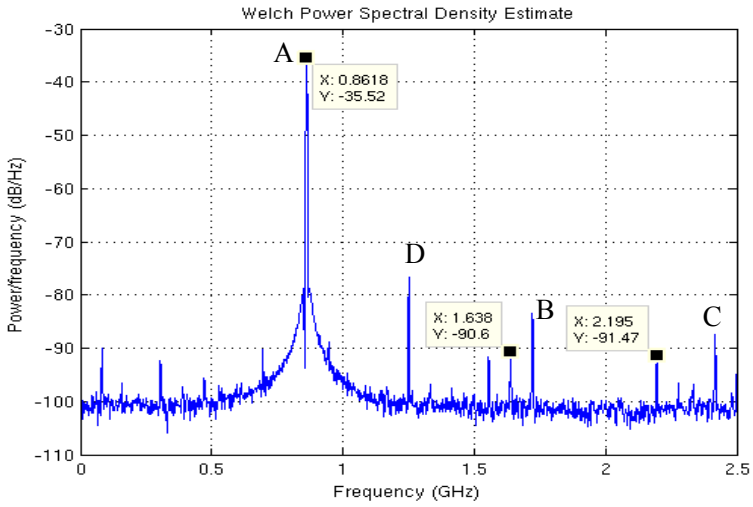
B. Gain Calibration

The gain is calibrated by feeding a sine wave into the ADC, and then estimating statistically the Root Mean Square (RMS) value of each core. By adjusting the gain in python using python software tools, the spurs in the noise background are suppressed. The signal is a sine wave (f_0 = 862 MHz) from signal generator. The data used for the plot are from the snapshot of the raw data at 5 GHz rate. Figure 3.11(a) shows the spectrum before gain adjustment. There are 9 prominent peaks (A-G) above -90 dB as marked in the figure. A is the original sine wave at 862 MHz. B and C are the second and third harmonics of the sine wave. (C originally located at 2586 MHz is aliased to the range of Nyquist frequency). D is at 1.25 GHz which each ADC core operate at and is due to the statistical properties of the 4 cores in ADC. E, F and G are the inter-productions of the sine wave and the different gains of the four cores. E is at 388 MHz which can be derived from 1.25 GHz subtract 862 MHz. F is at 1.638 MHz which can be derived from 1.25 GHz plus 862 MHz. G is at 2.112 GHz which can be derived from 2.5 GHz subtract 862 MHz.

Figure 3.11(b) shows the spectrum after the gain is adjusted. We can see that the inter-products at 388 MHz (E), 1.638 GHz (F), 2.112 GHz(G) are suppressed. The typical value (magnitude) of the raw data is 50, and SNR after the gain adjustment is more than 40dB as expected.



(a) Spectrum before calibrating the ADC gains



(b) Spectrum after calibrating the ADC gains

Figure 3.11. Spectrum before (a) and after (b) calibrating the ADC gains. In the test, an 861 MHz sine wave from signal generator are used as the input. The data used for the plot are from the snapshot of the raw data at 5GHz rate. Comparing these two spectra, the spurs at 0.3882 GHz, 1.638 GHz, 2.112 GHz are suppressed, which are the largest spurs (except the one at 1.25GHz introduced by the statistical properties of the 4 cores in ADC, and truncation of the ADC). The typical value (magnitude) of the raw data is 50, and SNR after the gain adjustment is more than 40dB as expected.

3.3 DIGITAL DOWN CONVERTER ARCHITECTURE

The 4 samplers are clocked at 2.5 GHz for each of the 4 antenna RF inputs. The FPGA receives the digitized samples from each sampler as 8 parallel data streams. Each of these streams has a data rate one eighth of the sampler clock, i.e. 312.5 MHz, which is identical to the FPGA clock rate, as illustrated in Figure 3.4.

After the FPGA receives the digitized samples from the sampler, the desired frequency band is translated to baseband as In Phase/Quadrature (I/Q) form by digital multiplication with finite precision approximations of sine and cosine waveforms. The digital downconverter, local oscillator, and multiplier must operate on the 2.5 GSPS samples in parallel for each antenna and each frequency band.

3.3.1 *Frequency resolution and dynamic range analysis*

The Direct digital synthesizer (Xilinx DDS compiler v4.0), or numerically controlled oscillator (NCO), is a Xilinx block. It produces the sinusoidal waveforms of both sine and cosine which is used to frequency translate signals of interest down to baseband in I/Q form.

DDS compiler supports a Spurious Free Dynamic Range (SFDR) from 18 dB to 150 dB by varying the bit precision of the sinusoidal [78]. The bit width of the output can be determined as follows [78]:

$$\text{SFDR} = 6 \times \text{Output Width} \tag{3.11}$$

Ideally, we would want as high SFDR as possible. However, high SFDR corresponds to large number of output bits and thus a high computation complexity in the multiplication during mixing. Therefore, for example, in most our designs, 60 dB SFDR is chosen because it exceeds the precision of the 8 bit samples. The corresponding output width is 10 bit, which is not too large for later multiplication operations.

The frequency resolution Δf of the synthesizer is a function of the clock frequency f_{clk} and the number of input bits B employed in the phase accumulator, i.e. the phase increase and phase offset in range [0,1]. The frequency resolution is determined by

$$\Delta f = \frac{f_{clk}}{2^B} \tag{3.12}$$

A finer frequency resolution provides better carrier frequency recovery ability, but on the other hand, requires more FPGA resource (more lookup tables to store all possible frequencies), and thus cause time closure more difficult. In most of our designs in which FPGA clock is set to 338 MHz for Simulink simulation, we designed the DDS frequency with 100 Hz resolution, which fit on to the FPGA board with time closure; meanwhile it is sufficient for signals centered at several hundred MHz (such as DTV signals) to recovery the accurate carrier frequency in a later stage. In the future, experiments with higher and lower resolutions can be carried out for different applications.

We can calculate the required number of bits B for f_{clk} of 338 MHz and Δf of 100 Hz:

$$f_{clk} = 338 \text{ MHz}$$
$$\text{B} = \left\lceil \log_2\left(\frac{f_{clk}}{\Delta \text{f}}\right) \right\rceil = \left\lceil \log_2\left(\frac{338MHz}{100\text{Hz}}\right) \right\rceil = \lceil 21.78 \rceil = 22 \tag{3.13}$$

where ⌈ ⌉ denotes the ceiling operator.

When the DDS compiler is set as 'programmable', it accepts frequency and phase offset from outside through the 'data' input. The frequency and phase offset has to be B bits wide, and are actually accepted as phase angle increment and phase angle offset in range [0, 1]. Assume the DDS generates the following sinusoidal wave:

$$y = \sin\left(2\pi f \frac{n}{f_{clk}} + \Delta\varphi\right) \qquad n = 0,1,2, \ldots \tag{3.14}$$

where y is the output, f_{clk} is the clock frequency, f is the oscillation frequency, and $\Delta\varphi$ is the phase offset.

The frequency angle increment 'p_inc' and offset 'p_off' are related to f and $\Delta\varphi$:

$$p_{\text{inc}} = \frac{f}{f_{clk}}$$
$$p_{\text{off}} = \frac{\Delta\varphi}{2\pi} \tag{3.15}$$
$$y = \sin(2\pi p_{\text{inc}} n + 2\pi p_{\text{off}}) \quad n = 0,1,2, \ldots$$

It's clear that $0 < p_{\text{off}}$, $p_{\text{inc}} < 1$, so the 'data' format is unsigned B bits with binary point at 0 relative to the MSB (most significant bit).

3.3.2 *Architecture of concatenated synthesizer*

Because the ADC sampling rate is 8 times the FPGA system clock, the down-conversion cannot be achieved with a single DDS block which can only operates at system clock and output a sinusoidal waveform with center frequency f_0 less than half of the system clock. A new synthesizer architecture is adopted by cascading multiple slow-clocked DDS. The eight parallel DDS blocks with appropriate phase offset would be identical to a single synthesizer clocked at 8 times of each DDS.

Figure 3.12 shows the structure of the new synthesizer. It's comprised of 8 DDS compilers, each with the same central frequency f_0 but a different phase offset φ. The new DDS output is 8 streams in parallel, and is denoted by y_i, where i is an integer. The new consequence y_i (i = 0, 1, 2, 3,….) represents the generated sine wave clocked at ADC clock, while each stream output, such as y_0, y_7, y_{15}, is the sine wave generated by an individual DDS compiler.

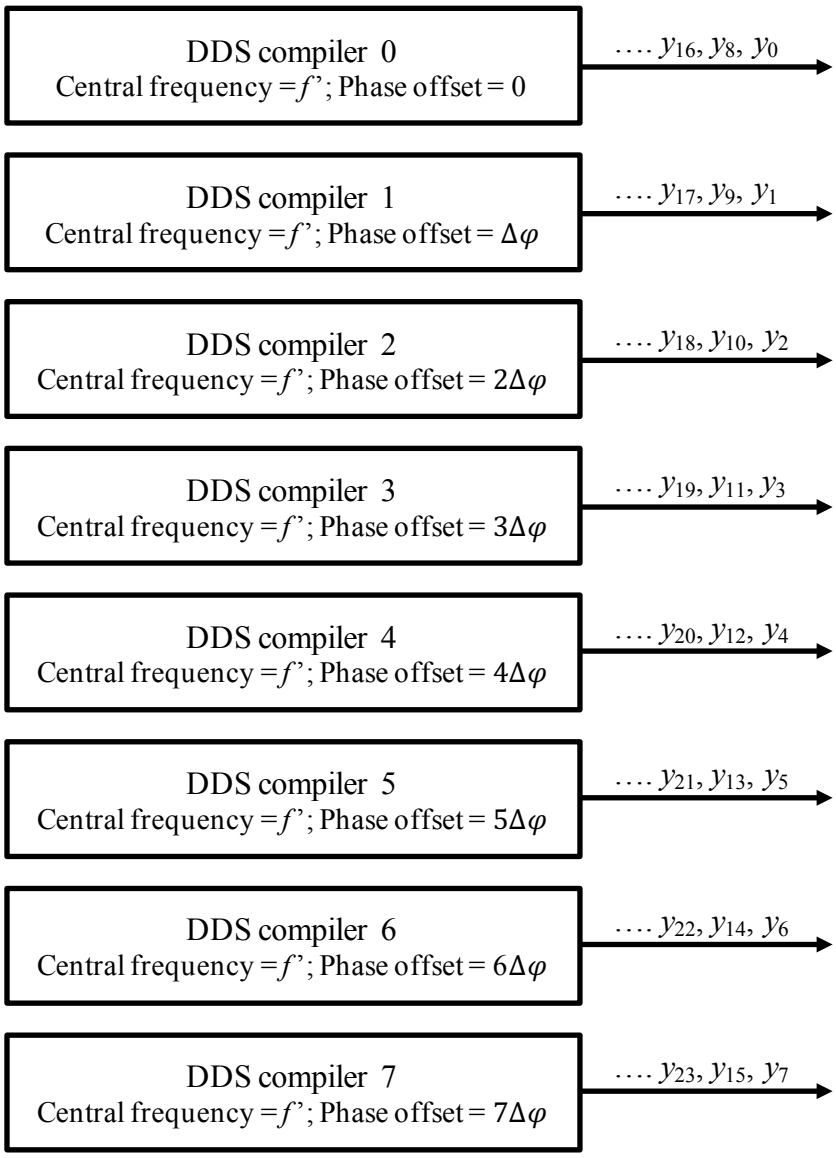


Figure 3.12. Structure of the new DDS generator clocked at 2704MHz. DDS compiler is a Xilinx block used to generate discrete the sine/cosine signals with a central frequency and initial phase. DDS compiler operates only at system clock which is 1/8 of the incoming sample rate in our design. Eight DDS compilers operate together with appropriate central frequency and phase offset settings and the 8 parallel output streams are concatenated to from a discrete sine signal that looks like operate 8 times faster than the individual DDS compiler output.

Now let's derive the values of f_0 and offset for each compiler, so that the new synthesizer could generate a discrete sinusoidal signal y_n with central frequency f_0 and ADC sampling rate f_s ($f_s = f_{adc} = 2704$ MHz).

$$y_n = \sin\left(2\pi f_0 \frac{n}{f_s}\right) \qquad n = 0,1,2,\ldots \tag{3.16}$$

First, let's look at the y_n spacing at 8 clock units:

$$\begin{aligned}
y_n &= \sin\left(2\pi f_0 \frac{n}{f_s}\right), \\
y_{n+8} &= \sin\left(2\pi f_0 \frac{n+8}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{1}{f_s/8}\right) \\
&= \sin\left(\phi_0 + 2\pi f_0 \frac{1}{f_{sys}}\right) \\
y_{n+8\times 2} &= \sin\left(2\pi f_0 \frac{n+16}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{2}{f_s/8}\right) \\
&= \sin\left(\phi_0 + 2\pi f_0 \frac{2}{f_{sys}}\right) \\
y_{n+8m} &= \sin\left(2\pi f_0 \frac{n+8m}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{m}{f_s/8}\right) \\
&= \sin\left(\phi_0 + 2\pi f_0 \frac{m}{f_{sys}}\right)
\end{aligned} \tag{3.17}$$

where f_{sys} is the system clock:

$$f_{sys} = \frac{f_s}{8} \tag{3.18}$$

Thus the new data consequence becomes a new sine wave discretized at a new clock frequency f_{sys}, 8 times slower than the original sampling rate f_s. The first term ϕ_0 in the sin() function is no more than a constant initial phase. Any f larger than f_{sys} would be aliased in to the range $[0, f_{sys}]$. Because the DDS compiler requires that the central frequency be positive, range $[0, f_{sys}]$ is chosen, instead of $[-f_{sys}/2, f_{sys}/2]$. Therefore, it's necessary to calculate a new center frequency f' that is the alias of f_0 with respect to f_{sys}:

$$f' = \mathrm{mod}(f_0, f_{\mathrm{sys}}) \tag{3.19}$$

where mod() is a modulus operator. Now f' is the central frequency of each of the 8 DDS compiler blocks.

The new data consequence of each DDS compiler now can be expressed as follows;

$$y_m = \sin\left(2\pi f' \frac{m}{f_{sys}}\right) \; m = 0,1,2 \ldots \tag{3.20}$$

Next, let's consider the phase offset of each DDS compiler. At time n, the 8 blocks should produce $y_n, y_{n+1}, y_{n+2}, y_{n+3}, y_{n+4}, .. y_{n+7}$, where

$$\begin{aligned}
y_n &= \sin\left(2\pi f_0 \frac{n}{f_s}\right), \\
y_{n+1} &= \sin\left(2\pi f_0 \frac{n+1}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{1}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + \Delta\varphi\right) \\
y_{n+2} &= \sin\left(2\pi f_0 \frac{n+2}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{2}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\Delta\varphi\right) \\
y_{n+3} &= \sin\left(2\pi f_0 \frac{n+3}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{3}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 3\Delta\varphi\right) \\
y_{n+4} &= \sin\left(2\pi f_0 \frac{n+4}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{4}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 4\Delta\varphi\right) \\
y_{n+5} &= \sin\left(2\pi f_0 \frac{n+5}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{5}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 5\Delta\varphi\right) \\
y_{n+6} &= \sin\left(2\pi f_0 \frac{n+6}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{6}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 6\Delta\varphi\right) \\
y_{n+7} &= \sin\left(2\pi f_0 \frac{n+7}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 2\pi f_0 \frac{7}{f_s}\right) = \sin\left(2\pi f_0 \frac{n}{f_s} + 7\Delta\varphi\right)
\end{aligned} \tag{3.21}$$

where

$$\Delta\varphi = \frac{2\pi f_0}{f_s} \tag{3.22}$$

$\Delta\varphi$ is the phase offset increment for each adjacent DDS compiler. For example, if the offset for compiler 0 is 0, the offset for the other 7 compilers are set to $\Delta\varphi$, $2\Delta\varphi$, $3\Delta\varphi$, $4\Delta\varphi$, $5\Delta\varphi$, $6\Delta\varphi$, $7\Delta\varphi$ in order.

The general form for each DDS compiler is

$$y_{m,n} = \sin\left(2\pi f' \frac{m}{f_{sys}} + n\Delta\varphi\right) \quad m = 0,1,2 \ldots \tag{3.23}$$

where *n* represents the *n*th DDS compiler.

In the previous simulation of generating 86 MHz and 1100 MHz signals, we can calculate the center frequency f' and offset increment for each signal based on Eq. 3.19 and Eq. 3.22. The center frequency f' of 1100 MHz signal is aliased to 86 MHz, so both signals have the same center frequency f'. The offset increment $\Delta\varphi$ is different for the two signals:

$$\begin{aligned} \Delta\varphi_{86MHz} &= \frac{2\pi f_0}{f_s} = \frac{2\pi 86}{2704} = 0.199835\, rad \\ \Delta\varphi_{1100MHz} &= \frac{2\pi f_0}{f_s} = \frac{2\pi 1100}{2704} = 2.55603\, rad \end{aligned} \tag{3.24}$$

The compiler also accepts the phase angle increment and phase angle offset, which are equivalent to central frequency f' and phase offset φ. They are connected with the relations：

$$y_{m,n} = \sin\left(2\pi f' \frac{m}{f_{sys}} + n\Delta\varphi\right) = \sin(2\pi \cdot phase_{inc} \cdot n + 2\pi \cdot \mathrm{phase}_{\mathrm{offset}}) \tag{3.25}$$

Here,

$$\begin{aligned} \mathrm{phase}_{\mathrm{inc}} &= \frac{f'}{f_{sys}} \\ \mathrm{phase}_{\mathrm{offset}} &= \frac{f_0}{f_s} \end{aligned} \tag{3.26}$$

For a programmable DDS configuration used in my design, the phase angle increment and phase angle offset are required and pre-calculated into the code. During FPGA compilation, a lookup table is generated based on the phase angle increment and phase angle offset, and it contains all possible center frequency values and the corresponding sinusoidal output value. The precisions of the values are specified by the input and output bit widths.

3.3.3 *DDS Simulation and Results*

This section examines the DDS block performance with Simulink simulation. First, the performance of a single DDS block is examined, and then the parallel DDS block is evaluated. By comparing the frequency spectrum of the single DDS block and the parallel DDS block, we can evaluate the degradation of SFDR.

First, a single frequency signal is generated from a single DDS block with specifications of f_0 = 86 MHz, $\Delta f = 100Hz$, $f_{clk} = 338\,MHz$, $\Delta\varphi = 0$, and 10-bit output. Figure 3.13 shows the spectrum of the signal. The signal is precise in frequency, with desired frequency peak and 50 dB SFDR.

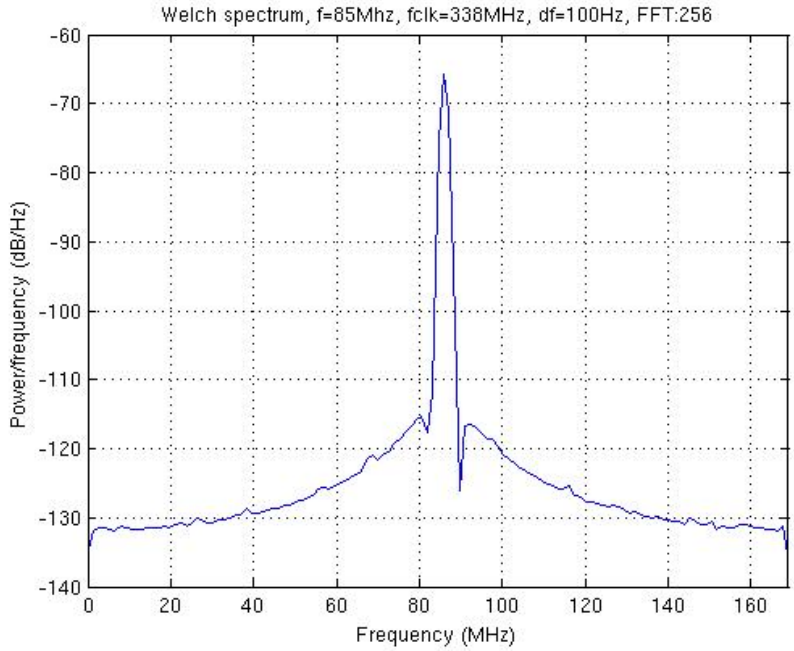


Figure 3.13. Spectrum of data generated from a single DDS block when $f_0 = 86\text{MHz}$, $\Delta\text{f} = 100\text{Hz}$, $f_{clk} = 338\,\text{MHz}$ and 10-bit output. Note that The signal is precise in frequency, with desired frequency peak and 50 dB SFDR.

Second, the eight parallel DDS blocks with appropriate phase offset is designed to have an equivalent synthesizer clocked at 8 times of the real DDS system clock, i.e. 2704MHz. Figure 3.14 and Figure 3.15 shows the spectrum of the concatenated synthesizer with center frequency f_0 at 86MHz and 1100MHz respectively. The 1100MHz synthesizer was generated with each DDS block having an aliased center frequency of 86MHz.

The noise floor of the synthesizer in Figure 3.14 is about 65 dB below the peak of the signal. The additional spurs above 1GHz degrade the SFDR, but the signal to spur ratio is about 50 dB, which is as same as the SFDR of a single DDS block in Figure 3.13.

As we have calculated in the last section, both synthesizers have the same aliased center frequency *f*' of 86MHz, but a different phase offset increment. The simulation further confirms the capability of the parallel synthesizer which can achieve a non-aliased frequency up to 2704/2MHz, as if it's a single synthesizer which operates at 2704 MHz clock rate.

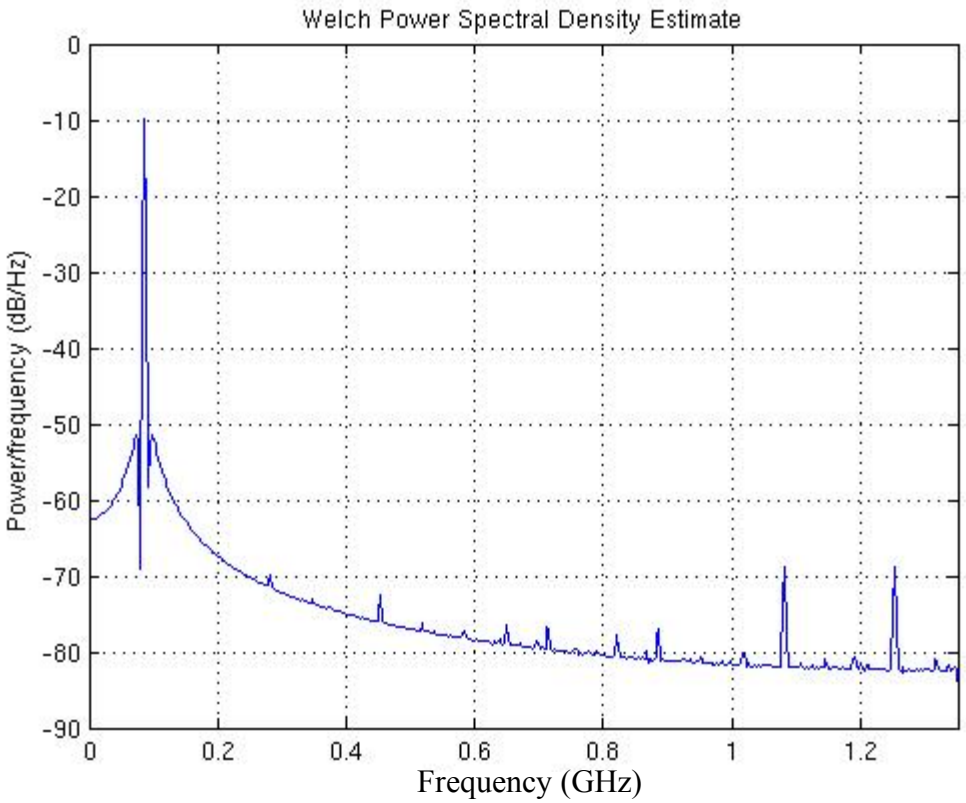


Figure 3.14. The spectrum of the 86 MHz sine wave generated from the eight parallel DDS compilers. Each individual DDS compiler has with system clock with a central frequency of 86 MHz and different phase offset settings. The eight stream output sine waveform signals are concatenated at each clock cycle and form a single sine wave with central frequency of at 86MHz and sample rate of 8 times system clock.

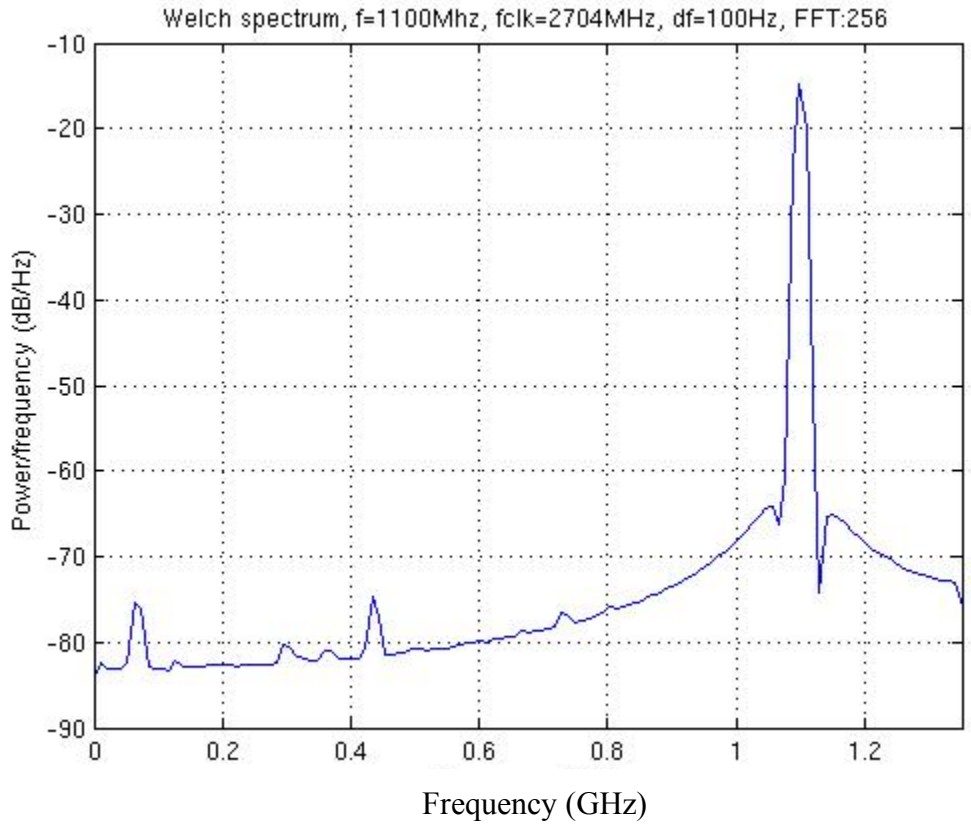


Figure 3.15. The spectrum of the 1100 MHz sine wave generated from eight parallel DDS compilers. Each individual DDS compiler has with system clock with a central frequency of 86 MHz and different phase offset settings. The eight stream output sine waveform signals are concatenated at each clock cycle and form a single sine wave with central frequency of at 1100 MHz and sample rate of 8 times system clock. Note that compare with Figure 3.14, each DDS compiler used to generate sine wave has the same central frequency but different phase offset.

3.3.4 *Control Logic Design for the Programmable DDS Compiler*

There are three levels to help understand the implementation of digital down conversion in FPGA. On the top level, first you need specify a new mixer frequency. Then in a lower level, you need to specify not only the frequency *f'* but also the frequency-dependent phase offset $\Delta\varphi$ for each of the eight parallel DDS compilers. The python code handles this level where these two values are are computed and send to the FPGA register (yellow blocks as shown in Figure 3.16. At the bottom level, a control logic need be designed as the Xilinx DDS can only accept either the frequency or phase offset at a time, not both at the same time. We discussed the computation of frequency and phase offset previously, in the section we will focus on the control logic design, that is, how to control DDS compiler such that during runtime it can be programmed correctly. Also as we mentioned that the DDS compiler will take in the frequency and phase offset in the form of phase angle increment and phase angle offset respectively when it's set to programmable, you would see the use of the phase angle increment and offset below, instead of frequency and phase offset.

A programmable DDS would provide flexibility of the frequency channel selection. Figure 3.16 shows control diagram of the programmable DDS compiler. If the DDS compiler is set programmable, the complier will, instead of using the fixed parameters, accept phase angle increment (phase_inc) and phase angle offset (phase_offset) from outside through port 'data'. Since the 'phase_inc' and 'phase_offset' are received through one port, then a second input 'reg_select' is used and serves as a MUX.

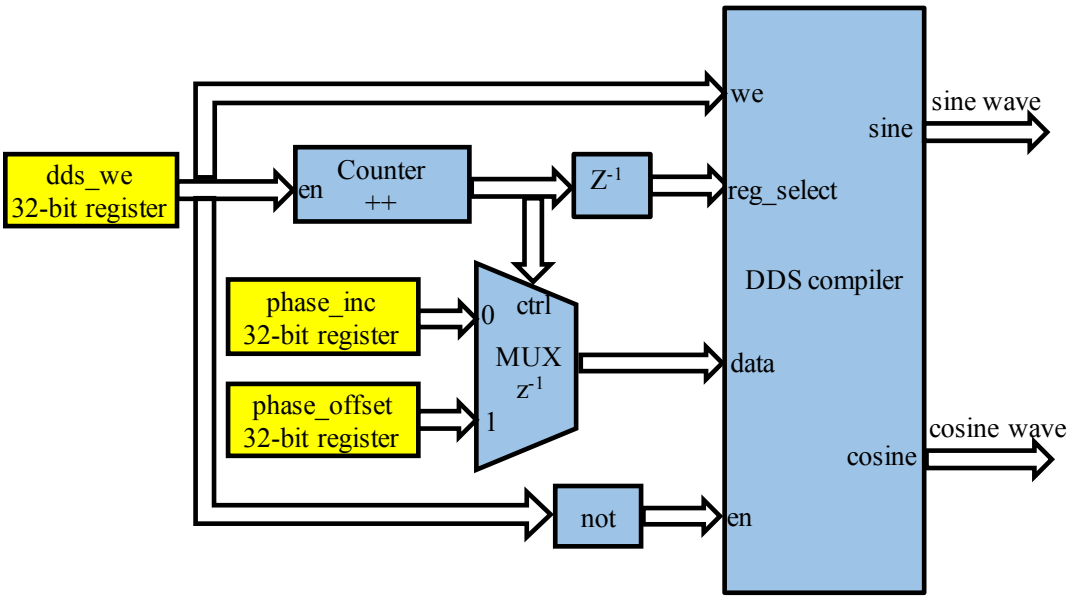


Figure 3.16. Control diagram for programmable DDS compiler. Users program the DDS compiler via the yellow blocks. The 'dds_we' block enables the DDS compiler start a new oscillator frequency, during which the compiler disables output, and takes in program parameters of phase_inc and phase_offset via 'data' port. As there is only one port to write two parameters, Boolean 'reg_select' port is used to control which address the data is written to. The Mux block is used to select which parameter is sent to 'data' port. The control logic block automatically synchronizes the 'reg_select' and 'data' ports, so that 'phase-inc" and 'phase_offset' can be written to the correct addresses.

The 'Data' port in the DDS compiler is connected to two registers where the phase_inc and phase_offset data are saved inside DDS compiler. It can be written during runtime. The 'reg_select' input controls which register to write into (when 'reg_select' is 1, 'data' port takes value into phase_offset register; when 'reg_select' is 0, 'data' port takes value into phase_inc register.) The 'we' port in DDS compiler is the write-enable port, which permits writing into the 'data' port when set to 1. The 'en' port is used to enable/disable DDS compiler to output sine wave.

The 'dds_we' signal in Figure 3.16 is a one-bit (boolean) signal. It's connected to the 'we' port in DDS compiler, so it's used to control when to write new value into DDS compiler to configure a mixer with a new central frequency. The 'we' port is also connected to 'en' port of a counter. When it is set to 1, it enables the counter in Figure 3.16 to start counting, and enables 'we'

port to accept new values for the ‘data’ port, and disabled the ‘en’ port of DDS compiler to temporarily pause generating sinusoidal waveform while configuring a new mixer.

The most significant bit of counter output is the input of ‘reg_select’, and it is also the control signal for the MUX which selects phase_inc or phase_offset. When the counter starts counting from all ‘0’s to ‘0111…111’, reg_select is 0, and the MUX automatically selects ‘phase_inc’, and then ‘phase_inc’ is sent to the correct register. When counter counts from ‘100000’ to all ‘1’s, ‘reg_select’ is 1, and the MUX automatically selects the ‘phase_offset’ value.

The automatic selection helps save time for the operator to manually select the correct bit for ‘reg_select’ and the corresponding values for ‘data’. It reduces the programming complexity on the operator side, and the chance of making mistakes of reading incorrect or invalid values into ‘data’.

Notice that there is one time-unit delay block between DDS compiler and ‘reg_select’. This is because the MUX takes one time-unit delay for computation, and the delay block ensure that the time that ‘phase_inc’ and ‘phase_offset’ are writing into DDS compiler are appropriately synchronized with the ‘reg_select’ signal.

Finally, four antennas of the same frequency band share parallel synthesizers, which helps synchronize signals from four antennas.

3.4 Polyphase Filter Architecture

3.4.1 *Polyphase Filter*

Polyphase decimation is achieved with a multi-rate filter bank. Decimation is the act of reducing the rate of a signal by first using an anti-alias filter and then downsampling.

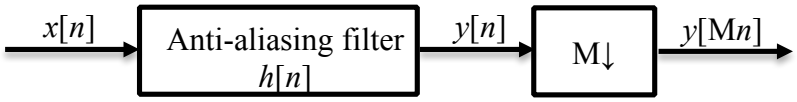


Figure 3.17. Block diagram of a traditional decimation. It involves an anti-aliasing filter and a sample rate decimation of M. The output *y* is M times slower than the input *x*.

Figure 3.17 shows a conventional decimation. First an anti-aliasing filter is applied to the incoming data at the high input data rate, and then one out of every M data of the filter output y[n]

is selected to achieve a lower output data rate (1/M of the input data rate). With a finite impulse response (FIR) filter of order N, it takes N multiplications and N-1 additions to compute each output sample $y[n]$. Therefore, M output samples requires M(2N-1) computations. Because only one of the M output samples are kept, this method is quite wasteful.

Polyphase decimation is a multi-rate filter bank that will only compute the values of $y[n]$ intended to keep. Therefore, it reduces the computation complexity by a factor of M. Moreover, it is implemented in a parallel way and the computation thus can be carried out at the speed of output rate, rather than the high input rate. This eases the clock rate for the FPGA design. For example, in our receiver, the incoming data for Polyphase filter is at a rate of 2.5 GHz or 5 GHz, and the output data rate is reduced to 312.5 MHz. The FPGA has no direct way to implement a FIR filter at 2.5GHz or 5GHz speed.

Here we won't go into details of the mathematic description of Polyphase filter, since it can be easily found in many places, such as [79]. The implementation can be easily understood with the block diagram in Figure 3.19. Take a decimation ratio M=3 for example. Figure 3.18(a) shows the conventional FIR filter implementation diagram, and its polyphase form is shown as Figure 3.18(b). First we divide the filter coefficients into M groups (M is also the decimation rate and is 3 in this example). The incoming data is also divided into M parallel streams, and each stream is then filtered by the corresponding sub filter bank. The output of each filter is summed up to produce the decimated output data $y[3n]$.

In our design, for the 2.5GSPS mode, each ADC outputs 8 parallel data streams with each at 312.5MHz rate. Therefore, the FIR filter coefficients is divided into 8 groups, i.e. 8 sub FIR filters. Each data stream is fed into the corresponding sub filter and then the 8 parallel samples out of the sub filters are summed up. Thus the incoming 8 parallel streams become one stream clocked at 312.5MHz, 8 times slower than ADC sampling rate.

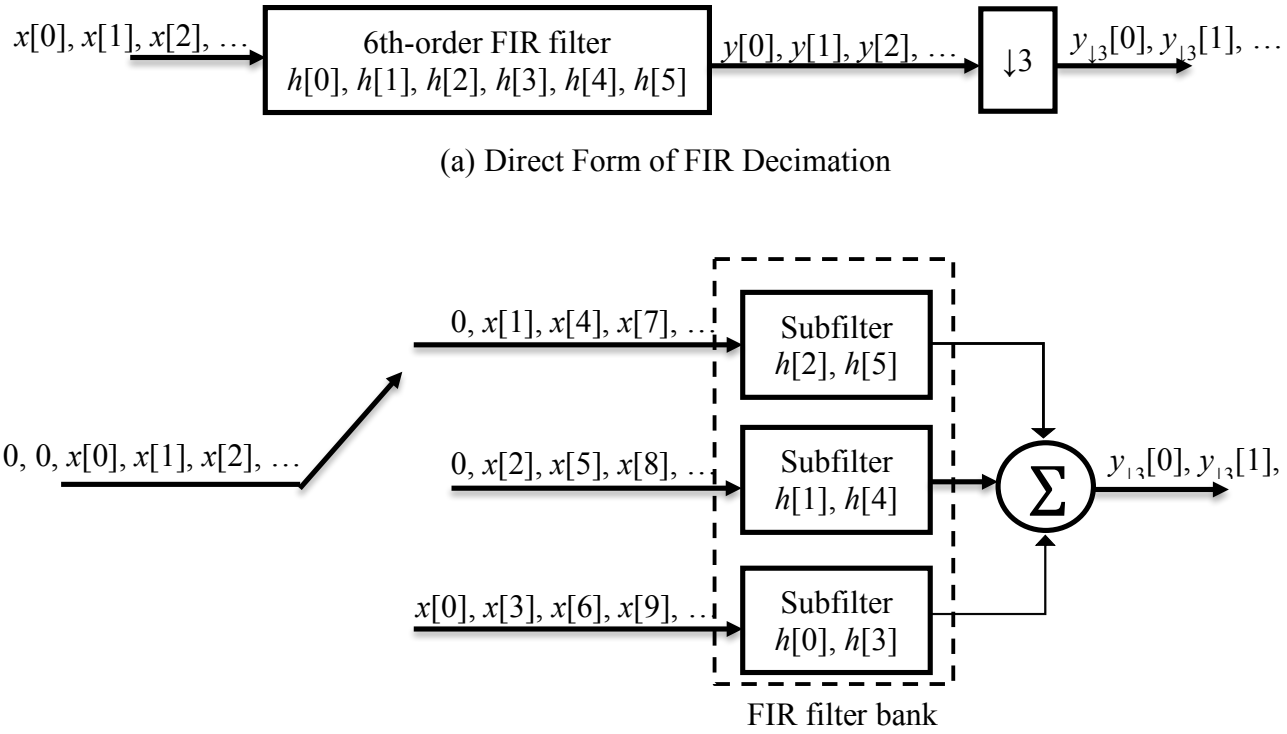
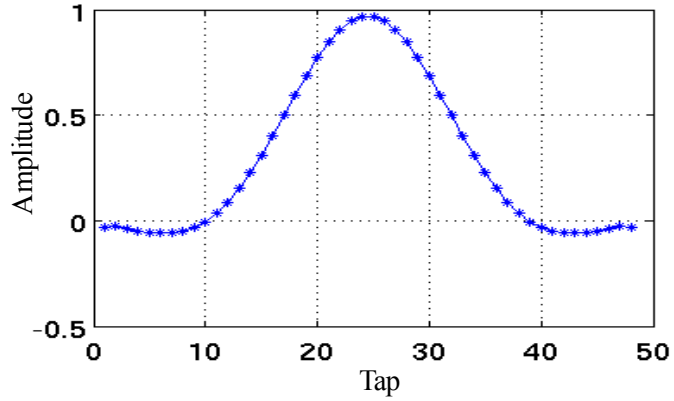


Figure 3.18. Example Polyphase filter implementation of width 3, implementing symmetric 6 order. The six filter coefficients are divided into 3 groups of sub filter bank. The incoming data is also divided into 3 parallel streams, and each stream is then filtered by the corresponding sub filter bank. The output of each filter is summed up to produce the decimated output data $y[3n]$

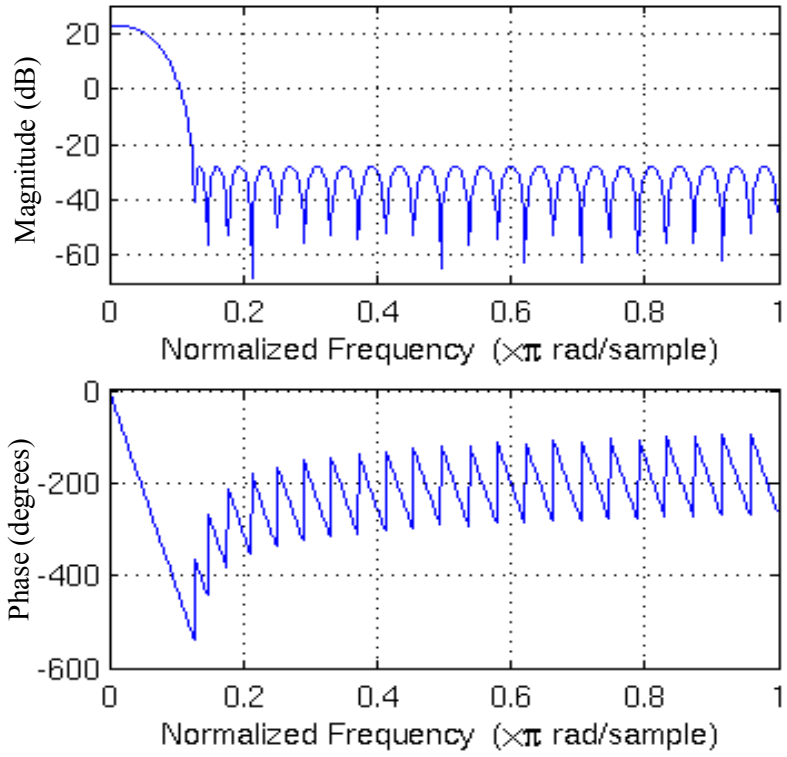
3.4.2 *Impulse Response and Frequency Response*

After the digital down conversion, a low-pass FIR filter is applied to the I/Q data in order to suppress unwanted frequency components and to decimate the data by 8 to become a serial signal synchronous with the FPGA clock. Because the data emerging from the DDS is still full rate, but parallel, an M=8 polyphase filter (PPF) is used for initial low-pass filtering followed by immediately decimation. This reduces the I/Q data rate to be synchronous with the FPGA clock at 312.5 MSPS; note that this is the highest possible output data rate for this digital receiver architecture. The PPF has 48 coefficients with 16-bit precision, as shown in Figure 3.19(a). It is symmetric about the 24th and 25th taps. The values of the last 24 taps are listed in Table 3.1. The

Polyphase filter provides about 10 dB processing gain. Figure 3.19(b) shows the frequency response of PPF. It has 50 dB attenuation at the stop band.



(a) Coefficients of the PPF



(b) Frequency response of PPF

Figure 3.19. Impulse response and frequency response of the Polyphase filter (PPF). The filter has 48 filter coefficients, and a cutoff frequency of 1/8 of the Nyquist frequency.

Table 3.1. PPF coefficients (48th order FIR filter with symmetry) The tap coefficients have 16-bit resolution.

Tap number	Amplitude	Tap number	Amplitude
25	0.965	37	0.087
26	0.944	38	0.035
27	0.904	39	-0.004
28	0.845	40	-0.032
29	0.772	41	-0.049
30	0.687	42	-0.058
31	0.594	43	-0.059
32	0.498	44	-0.055
33	0.402	45	-0.047
34	0.311	46	-0.038
35	0.226	47	-0.028
36	0.151	48	-0.032

3.4.3 *The Alternative Approach for Achieving Down-Conversion and Decimation*

The alternative approach for achieving down-conversion and decimation is to apply a band pass filter at required frequency first and then convert the frequency band down to baseband. The structure of DDC with a LP polyphase filter outweighs the structure of bandpass filter followed with DDC based on the engineering considerations related to the FPGA resources and aliasing issue. The polyphase module comes with filtering and decimation together, and therefore, has a unaliased range up to half of the FPGA clock rate. If the polyphase filter comes first before DDC, then any required frequency beyond unaliased range is folded, therefore, not only the polyphase filter has to be reloadable or be designed separately for different frequency range, and the aliasing has to be considered when frequency of the DDC is calculated. The reloadable filter coefficients consume a large amount FPGA RAM resources besides the slices used for the reloading logic. Furthermore, the band pass filter has much more coefficients than a low pass filter, and thus required more multiplication/add operations. The current DDC followed by polyphase filter is much easier and more efficient in terms of design and operation.

3.5 CIC FILTER ARCHITECTURE

After the polyphse filter, the signal bandwidth is decimated to 312.5 MHz which is still too wide compared to the DTV bandwidth, and also is difficult for 10Gb Ethernet port to catch up. We thus need further reduce the data rate to the DTV bandwidth. A decimation ratio of 30 is required to reduce date rate from 312.5 MHz to around 10 MHz. We of course achieve it with a lowpass FIR filter. However, a traditional FIR filter with this decimation requires a large number of coefficients and thus the FPGA DSP48E1 slices which are designed for high speed DSP operations, can't be performed with FIR filter only because of the limited availability of the 2046 FPGA DSP48E1 slices. The DSP48E1 slices required is proportional to the length of the filter coefficients. For example, a FIR filter with decimation of 8 is 48 long, and it uses up 48 DSP48E1 slices. Consider the IQ forms and 4 antenna paths, the total slices used for one frequency band is 48x2x4= 384. If the decimation is 30, the length of the filter coefficient vector would be several hundred which is beyond the FGPA's capacity. Therefore, CIC filter is introduced to achieve the second decimation with much less computation complexity.

3.5.1 *CIC Filter and Implementation*

The Cascade Integrator Comb (CIC) filter is known for narrow band filtering with much less computation complexity than any FIR filters. The CIC filter only uses addition and subtraction operations, which reduces the computation complexity. It also saves the memory ordinarily required to store filter coefficients as a regular FIR filter requires, especially when the decimation rate R is large. Also since it doesn't have any multiplication or division operations, the computation can be much faster, which benefits timing in the FPGA logic, the main concern since the FPGA operates at high clock rate. Finally, the CIC filter can be easily achieved without a DSP block.

With the benefit of the programmable property of the frequency of DDS compiler and decimation of CIC compiler, we can control the frequency and bandwidth during run time by carefully designing the logic to write the frequency and decimation appropriately to the related FPGA registers. The buffer of the Ethernet transmission is also well designed to read in data at the appropriate rate and read in header information for each UDP packet.

The following description of the CIC decimator and is based closely on that provided by [80]. The general concept of a CIC filter is the low-pass response that results from filtering an input signal with a cascade of N unit-amplitude, rectangular windows of length RM, where R is the rate change (decimation), and M is the differential delay in the comb section stages of the filter. N is also called the number of CIC stages. The system response of such filter is

$$H(z) = \left[\sum_{k=0}^{MR-1} z^{-k}\right]^{N} \tag{3.27}$$

or

$$H(z) = \frac{(1 - z^{-M \cdot R})^{N}}{(1 - z^{-1})^{N}} \tag{3.28}$$

The implementation of this filter response with a clever combination of comb filter sections, integrator sections, and up-sampling (for interpolation) and down-sampling (for decimation) gives rise to the hardware-efficient implementation of CIC filters.

A block diagram of realization of this response can be seen in Figure 3.20. There are two sections to the CIC decimator filter: an integrator section with *N* integrator stages that processes input data samples at a sampling rate f_s, and a comb section that operates at the lower sampling rate f_s/R. This comb section consists of *N* comb stages with a differential delay of *M* samples per stage. The down sampling operation decimates the output of the integrator section by passing only every Rth sample to the comb section of the filter.

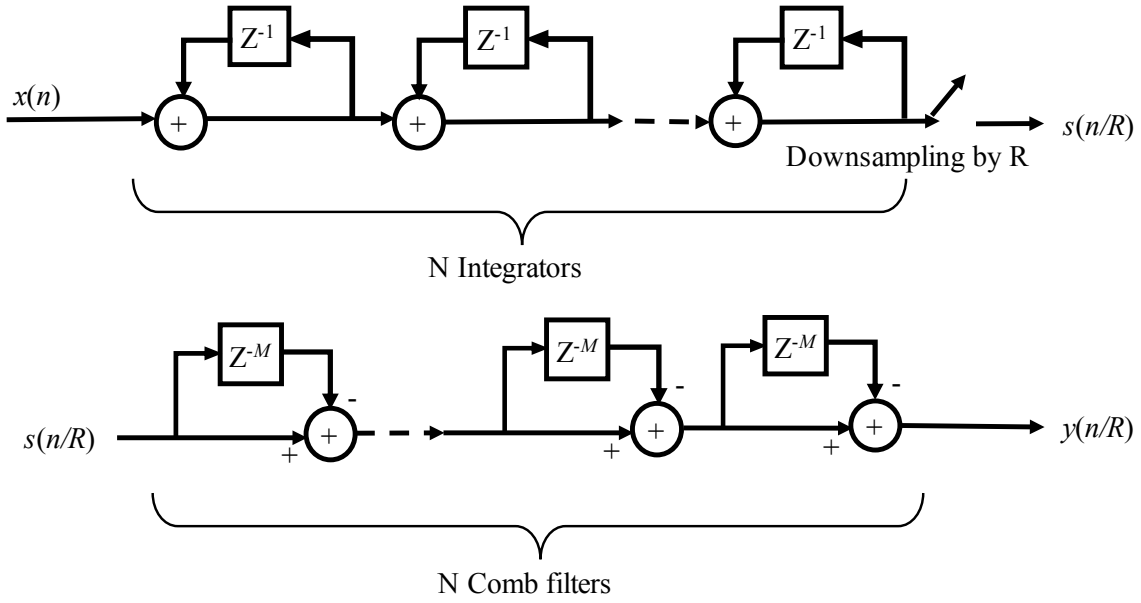


Figure 3.20. Block diagram of CIC decimation filter. It has an integration section of N integrators and a comb section of N comb filters with a decimated rate of R.

We then derive the magnitude frequency response of CIC filter from Eq. 3.28

$$H(z) = \left[\frac{\sin(\pi RMf)}{\sin(\pi f)}\right]^N \tag{3.29}$$

where f is the discrete-time frequency, normalized to the higher frequency in a rate changing filter (the input sampling frequency in a CIC decimation). This magnitude response is low-pass. In the design process of a CIC filter implementation, the parameters R, M, and N are selected to provide adequate passband characteristics over the frequency range from zero to a predetermined cutoff frequency f_c. Figure 3.21 shows the frequency response of a 6-stage (N = 6) CIC filter with unity differential delay (M = 1) and a sample rate change R = 10.

3.5.2 *CIC Filter Compiler and Modeling*

In our design, we applied a 6-stage (N = 6) CIC filter with unity differential delay (M = 1) and a programmable sample rate change in the range R∈[4, 50]. It is achieved with the Xilinx CIC

compiler. Figure 3.21 and Figure 3.22 shows the simulation results (Frequency and impulse responses) when *R* is set to 10.

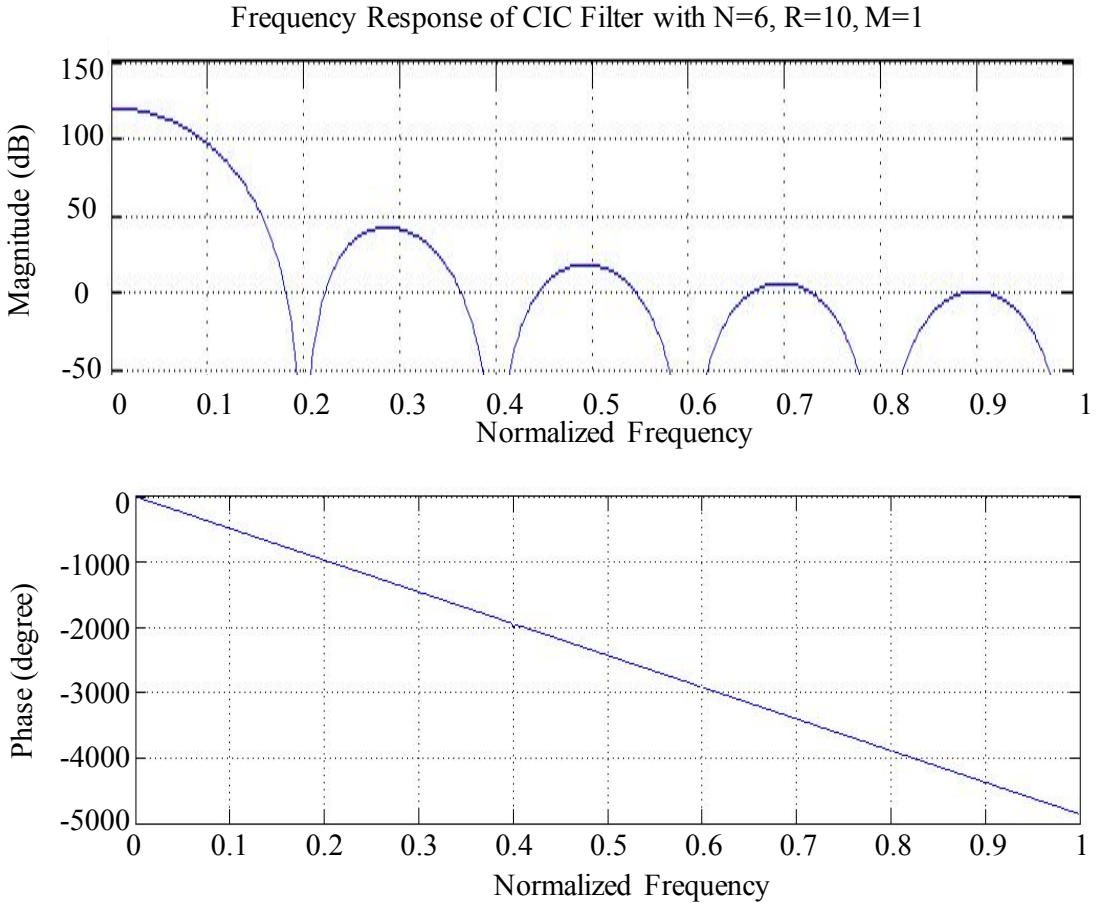


Figure 3.21. Frequency Response of the 6-stage (*N* = 6) CIC filter with unity differential delay (*M* = 1) and a sample rate change of 10 (R=10). note that the group delay is (1000/180xpi)/(0.2x2/fs)=28/fs seconds or 28 sample clock cycles.

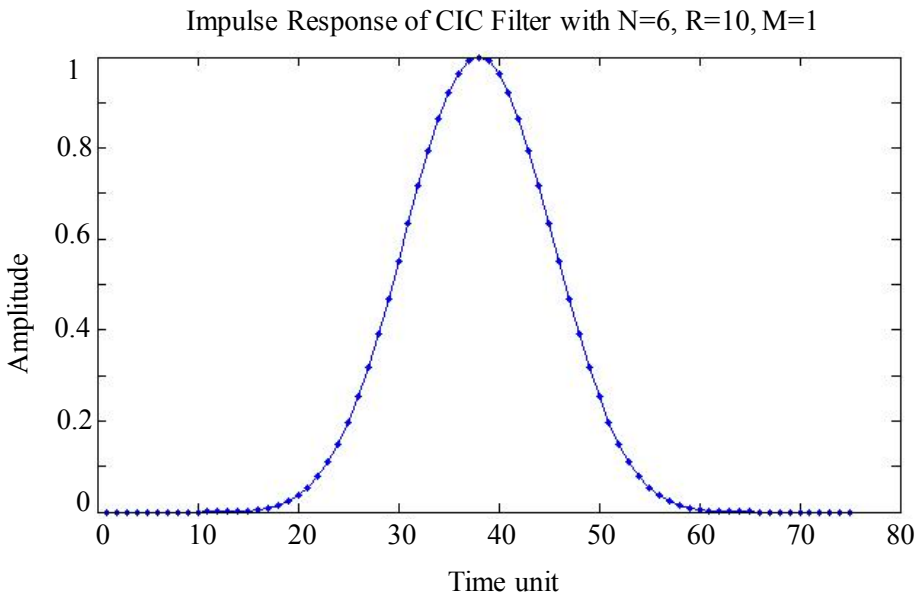


Figure 3.22. Impulse response of the CIC filter in Figure 3.21. Its quasi Gaussian shape reveals its good low pass response.

Note that the CIC Xilinx block requires a minimum decimation rate of 4. Therefore, the bandwidth after the CIC filter is at most 312.5 MHz/4 = 78.125 MHz. Another feature of the CIC filter is that the adjacent stop band is not ideally attenuated. After decimation, the effective bandwidth is only 80% of the baseband, and 20% spectrum affected by aliasing happens at the edge of the cutoff frequency. The adjacent spectrum is not suppressed very well and thus folded into the edge. Care was taken to ensure that the CIC decimator parameters are properly chosen to avoid detrimental effects from aliasing.

3.5.3 *Register Growth in CIC Decimator*

The CIC datapath undergoes internal register growth that is a function of all the design parameters: *N, M, R* in addition to the input sample precision *B*. As shown in Eq. 3.30, the output bit width of a CIC decimator with full precision is given by

$$B_{max} = \lceil N \log_2 RM + B \rceil \tag{3.30}$$

where $\lceil\ \rceil$ denotes the ceiling operator. The CIC Compiler supports both full and limited precision output. For full precision, the CIC decimator implementation uses B_{max} bits internally for each of

the integrator and differentiator stages. This introduces no quantization error at the output. For limited precision (that is, output bit width less than B_{max}), the registers in the integrator and comb stages are sized to limit the quantization noise variance at the output as described in [80]. Consequently, the hardware resources in a CIC decimator implementation can be reduced when using limited precision output at the cost of quantization noise. This ability to trade off resources and quantization noise is important to achieve an optimum implementation.

Our designs have the maximum of the programmable R of 50, N=6, M=1, input bit width of 18. The bitwidth of the output data is

$$B_{max} = \lceil N \log_2 RM + B \rceil = \lceil 6 \log_2 (50 \times 1) + 18 \rceil = 52 \qquad (3.31)$$

where $\lceil \quad \rceil$ is the ceiling operator. This is typically truncated to 8 bits or 16 bits later for transmission through 10GbE network.

3.5.4 *Control Logic Block Design for Programmable CIC Compiler*

The Xilinx CIC filter can be programmable. It can be programmed to change the decimation rate R during run time. The control logic is similar but simpler than that of DDS compiler. Unlike two parameters ('phase_inc' and 'phase_off') which are needed for the DDS compiler, CIC compiler only has one parameter R, so the input ports contain only 'we', 'en' and 'data', without 'reg_select'. The control logic is similar to but simpler than the control logic for mixer, so here we won't go into details of the logic control.

Note that even though the CIC filter could decimate the data rate by R, the output rate is still kept at FPGA clock rate. It works in such a way that each decimated sample is repeated R times, from which 1 of R samples is selected later for transport off the FPGA. This may appear wasteful, but there is no way to redeploy the FPGA fabric.

With the benefit of the programmable property of the frequency of DDS compiler and decimation of CIC compiler, we can control the frequency and bandwidth during run time by carefully designing the logic to write the frequency and decimation appropriately to the related FPGA registers. The design of Ethernet transmission is also customized to accommodate the data rate.

3.5.5 *Automatic Bit Selection Block Design*

After the CIC filter, the samples in each antenna stream have grown to over 50 bits. The data is then scaled and truncated to retain the 8 most significant bits, and then these 8 bit samples for the 4 antennas are packed into a 64-bit word. The data typically have 10-12 bits of precision at this point, so some precision is sacrificed in order to fit into the available data bandwidth. When the 10 GbE buffer has filled it is transmitted to the network as a UDP packet; typical packet sizes are 4096 bytes or 8192 bytes. We have also experimented with a mode that preserves 16 bit precision, however this halves the available output data rate. In this section, we focus on the strategy to find out the most significant 8 successive bits.

2 MUX blocks are used in order as shown in Figure 3.23. The user can select the scale level by controlling 'sel' port on the two MUXs during run time. The signal was first reinterpreted with 4 binary points from 0 binary point, and is then divided into 4 streams scaled by 2^0, 2^{10}, 2^{20}, 2^{30} with 2^{10} increment. The selected stream with the first MUX is then further divided by 10 streams scaled by 2^0, 2^1, 2^2, …, 2^9 with 2^1 increment. Therefore, the combination of the 2 MUXs provides a continuous tuning range $[2^0, 2^{39}]$.

The ADD blocks function as rounding operation. The signal is added with a constant that only has a binary 1 on the rounding digit, and is then truncated, which mathematically is equivalent to a rounding operation. The output of the first scale-mux retains 3 binary points, indicating a constant of 2^{-4} (or -0.625). The output of the second scale-mux retains 0 binary point, indicating a constant of 2^{-1} (or 0.5). The reason of using ADD block instead of direct rounding block is to close timing problem met during compilation. We suspect rounding operation consumes much more hardware than adding operation.

The input has a total bit width of 50 with 46 integer digits and 4 fractional digits. The output of the second MUX has 8 integer digits. With the largest tuning range, the output would have 46-39=7 integer digits. The output could well reserve the most significant eight bits if the original signal occupies the most significant bit of the total 50 bits. On the other hand, as we can observe from Figure 3.23, the binary points are discarded during the second round scaling. Therefore, the smallest signal we can retain without signal quality degradation would be a signal occupying at least 8 integer digits. With the smallest tuning range (2^0), this signal would pass through this module with sufficient SNR.

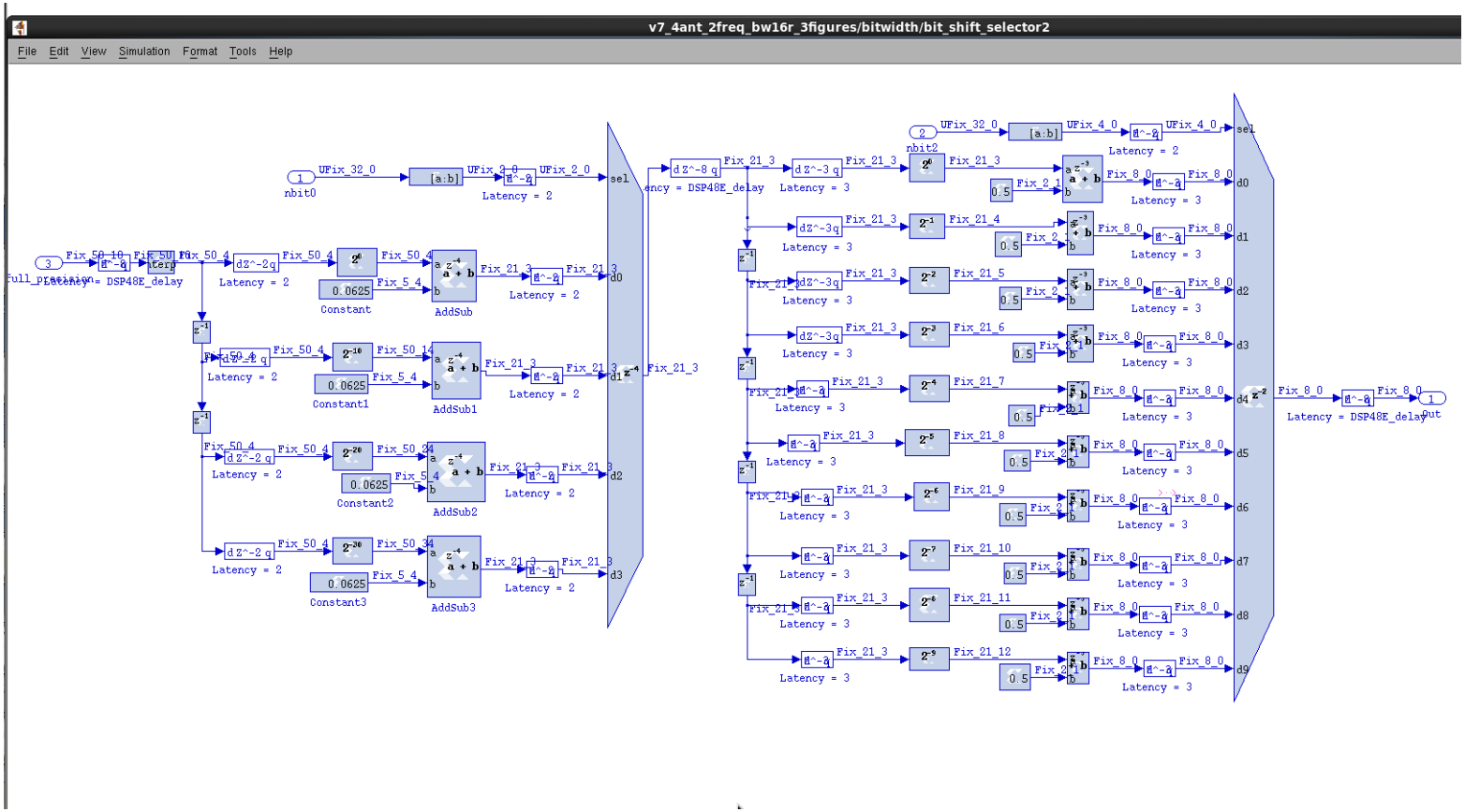


Figure 3.23. Automatic bit selection module. User can select the scale level by controlling 'sel' port on the two MUXs during run time. The signal was first reinterpreted with 4 binary points from 0 binary point, and is then divided into 4 streams scaled by 2^0, 2^{10}, 2^{20}, 2^{30} with 2^{10} increment. The selected stream with the first MUX is then further divided by 10 streams scaled by 2^0, 2^1, 2^2, …, 2^9 with 2^1 increment. Therefore, the combination of the 2 MUXs provides a continues tuning range $[2^0, 2^{39}]$.

3.5.6 *CIC Compensation Filter*

As the order of the zeros increase, the passband droop also increases, thus narrowing the filter bandwidth. The increased droop might not be acceptable in some applications. The droop is frequently corrected using an additional (non-CIC-based) stage of filtering after the CIC decimator. A compensation filter (not part of the CIC Compiler) can be used to flatten the passband frequency response.

For a CIC decimator, the compensation filter operates at the decimated sample rate. To achieve a flat pass band, the compensation FIR filter should have a magnitude response that is the inverse of Eq. 3.29, as shown in Eq. 3.32

$$G(z) = \left| MR \frac{\sin(\pi f/R)}{\sin(\pi Mf)} \right|^N \xrightarrow{if\ (\frac{\pi f}{R} \ll 1)} \left| \frac{\pi Mf}{\sin(\pi Mf)} \right|^N = |\mathrm{sinc}^{-1}(Mf)|^N \qquad (3.32)$$

When R is large, the compensation filter response can be approximated by the inverse sinc() function, so the compensation filter is sometimes referred to as the "inverse sinc filter." The compensation filter provides $[x/\sin(x)]^N$ shaping.

For our design, we didn't use CIC compensation filter for several reasons. First, CIC decimation factor R is usually 25, corresponding to the decimated data rate of 12.5 MHz. The DTV signal of interest is 6 MHz wide, which is less than half of the Nyquist sampling rate. Therefore, the passband for DTV signal is almost flat compared to the effect of frequency selective fading. Moreover, the aliasing at the CIC filter frequency edge is overcome by Nyquist filter used in DTV demodulation. Finally, the inverse sinc filter can be achieved subsequently in the GPU if needed.

3.5.7 *The Order of CIC Filter and Polyphase Filter*

The order of the polyphase FIR phase filter and CIC filter cannot be swapped, or replaced with each other. On one hand, although CIC filter is better to deal with large decimation, but it can only work at FPGA clock speed or lower; there is no polyphase form of the CIC filter due to the IIR filter component of the CIC filter, therefore, it's impossible to apply the CIC filter directly to the 2.5GHz data stream. Instead, the initial decimation of the 2.5 GHz data stream by 8 is necessary before the CIC filter can be applied.

3.6 GBE INTERFACE DESIGN

3.6.1 *GbE Yellow Block*

The output after bit selection is 8 bit for each I/Q channel and each of the four antennas. Therefore, there are total 8 streams of 8-bit samples for one frequency band in 2.5 GSPS mode. The 8 streams are combined to form a stream of 64-bit word and then sent to the GbE port of the GbE yellow block. The GbE yellow block is a Simulink block developed by CASPER community, and it contains the configuration of 10 Gb Ethernet interface of ROACH-2 board. The GbE yellow block adopts UDP for data transmission. UDP transmission is more efficient than TCP transmission as it saves the setup time for a reliable connection; and also as there are only two host PCs (data-receiving computer and ROACH-2 board) within the network, it is unlikely to have unreliable connections or data transmission conflict. As you will see later that the header added to each data packet provides capability to monitor packet order and packet drop during transmission.

A 64-bit word is required as the data format. The maximum GbE buffer size is [8k+512] words. Figure 3.24 shows the GbE yellow block and its peripheral connections. 'tx_data' is used to receive data. To operate GbE appropriately, we need to design the length of each data package and when to enable input port to acept data. These two parameters are configured by ports 'tx_valid' and 'tx_end_of_frame' of GbE yellow block.

Figure 3.25 shows the GbE yellow block parameter settings. Unlike Ethernet interfaces for PC that are usually set up dynamically in Linux or Windows operation systems, the 10Gb Ethernet interface of ROACH-2 is set up in the Simulink design, and will be fixed once compiled. Two important parameters are required to set up the interface correctly: the fabric MAC address and fabric IP address as shown in Figure 3.25. The destination IP address and UDP port, as shown on the upper right corner of Figure 3.24, are set to registers appropriately so that they can be specified by the operator during run time. The Xilinx MUX block (in light blue) in Figure 3.24 is used to insert header into the data stream. At the clock cycle when GbE yellow block starts buffering a new data packet, the MUX selects the 64-bit header word and send it to 'data' port of GbE yellow block. Then starting from the next clock cycle, the signal data is selected and sent to GbE yellow block. MUX is controlled by 'header_flag' bool variable. It may seem that we lose some data that are produced during the time header is transmitted. However, as CIC filter repeats every output

data by R times or R clock cycles (R is the decimation rate), we can in principle insert R-1 header words between two adjacent new signal data.

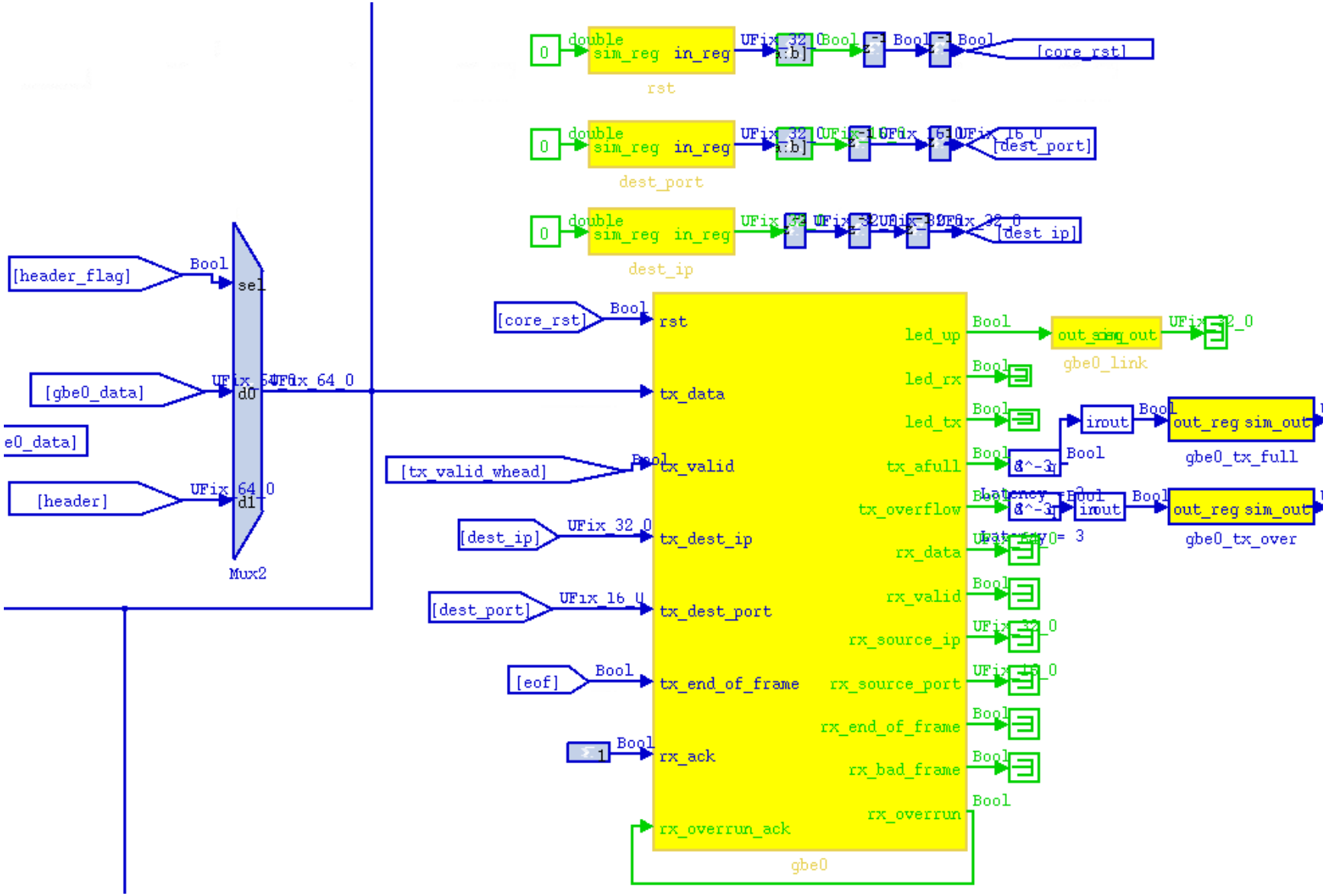


Figure 3.24. GbE yellow block and peripheral connections.

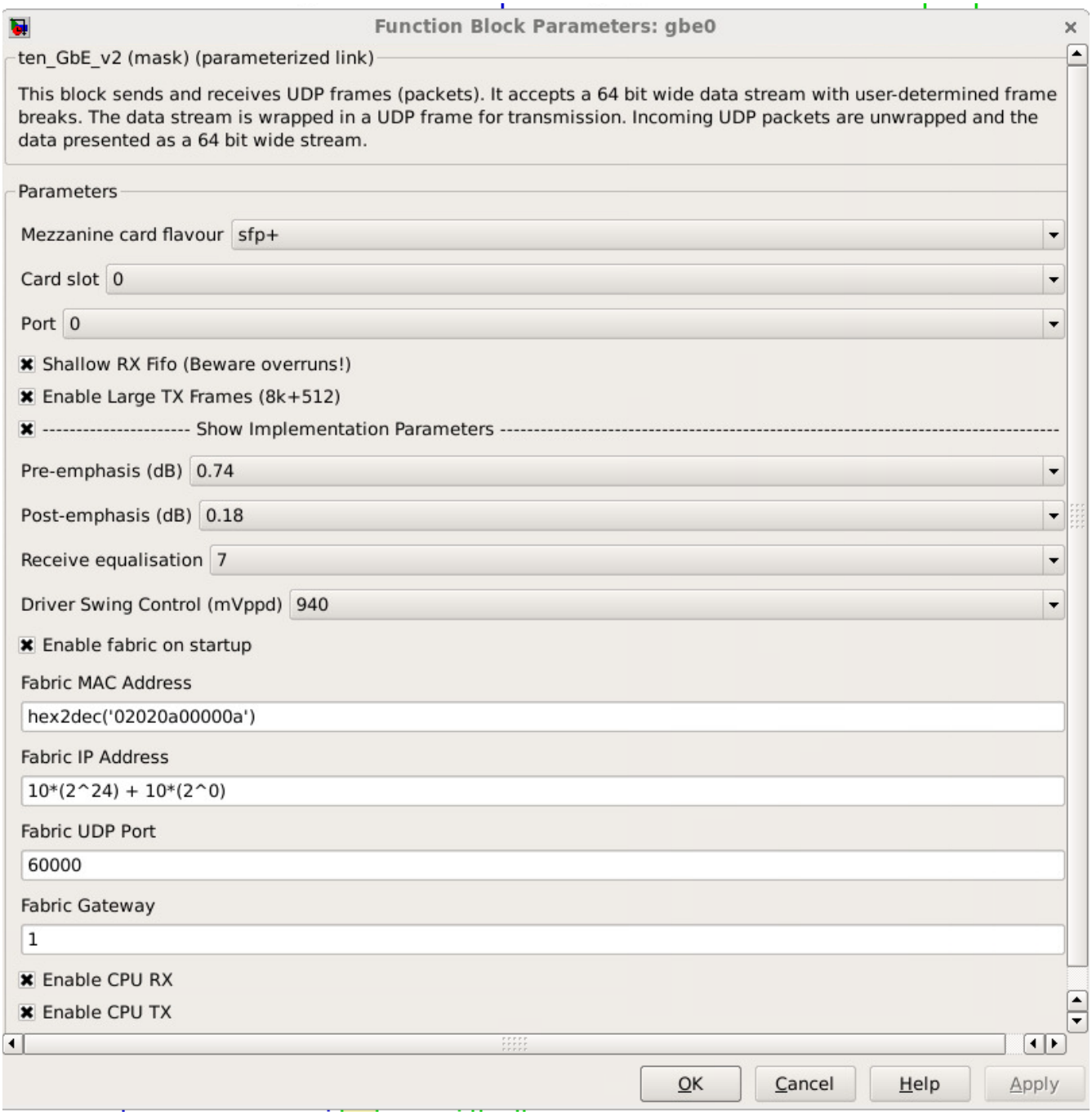


Figure 3.25. GbE yellow block parameter settings.

3.6.2 *GbE Control Logic*

Figure 3.26 shows the control logic subsystem for GbE yellow block. Figure 3.27 shows the inside of the control logic subsystem of Figure 3.26, including header composition.

The transmission rate of the 10GbE interface is automatically adjusted with the CIC decimation rate. when the CIC decimation rate changes, the output rate of CIC filter remains at FPGA clock rate, but the CIC filter has an output port that indicates the decimation rate. The GbE control logic would make adjustment based on this CIC complier output signal.

The 'pkt_count' block is used to generate control parameters for GbE yellow block, such as the 'tx_valid_w_header' signal for 'tx_valid' port, 'eof' signal for 'end_of_packet' port, etc, as shown in Figure 3.24).

As we discussed before, the output rate of CIC filter is at FPGA clock rate but the each decimated sample is repeated R times to keep the clock rate. Therefore, we don't need to send all samples off the board, but only send every 1 out of R samples.

The fact that the throughout rate can be adjusted automatically benefit from the CIC filter output 'DataRd', a bool signal that outputs 1 every other R samples and 0 for the rest of the time. Therefore, 'DataRd' can be used as the enable port ('tx_valid') that enable/disable the current sample write into the GbE buffer. A bool signal ('eof') is also required as the rising edge of 'eof' signal triggers an end to the current packet and a start to a new packet. In order to have a manageable packet size, the 'eof' signal need to be carefully designed. The 'eof' signal is generated in the 'pkt_count' subsystem as shown in Figure 3.27. An accumulator block is used to count the number of 'tx_valid' pulse which represents the number of samples read into the GbE buffer. Once it counts up to the packet size specified in the run time, a unit pulse is generated and the accumulator is reset to 0 to start a new accumulation.

With the benefit of this logic, we can see that the package size can be adjusted during run time as well, which offers flexibility for network transmissions.

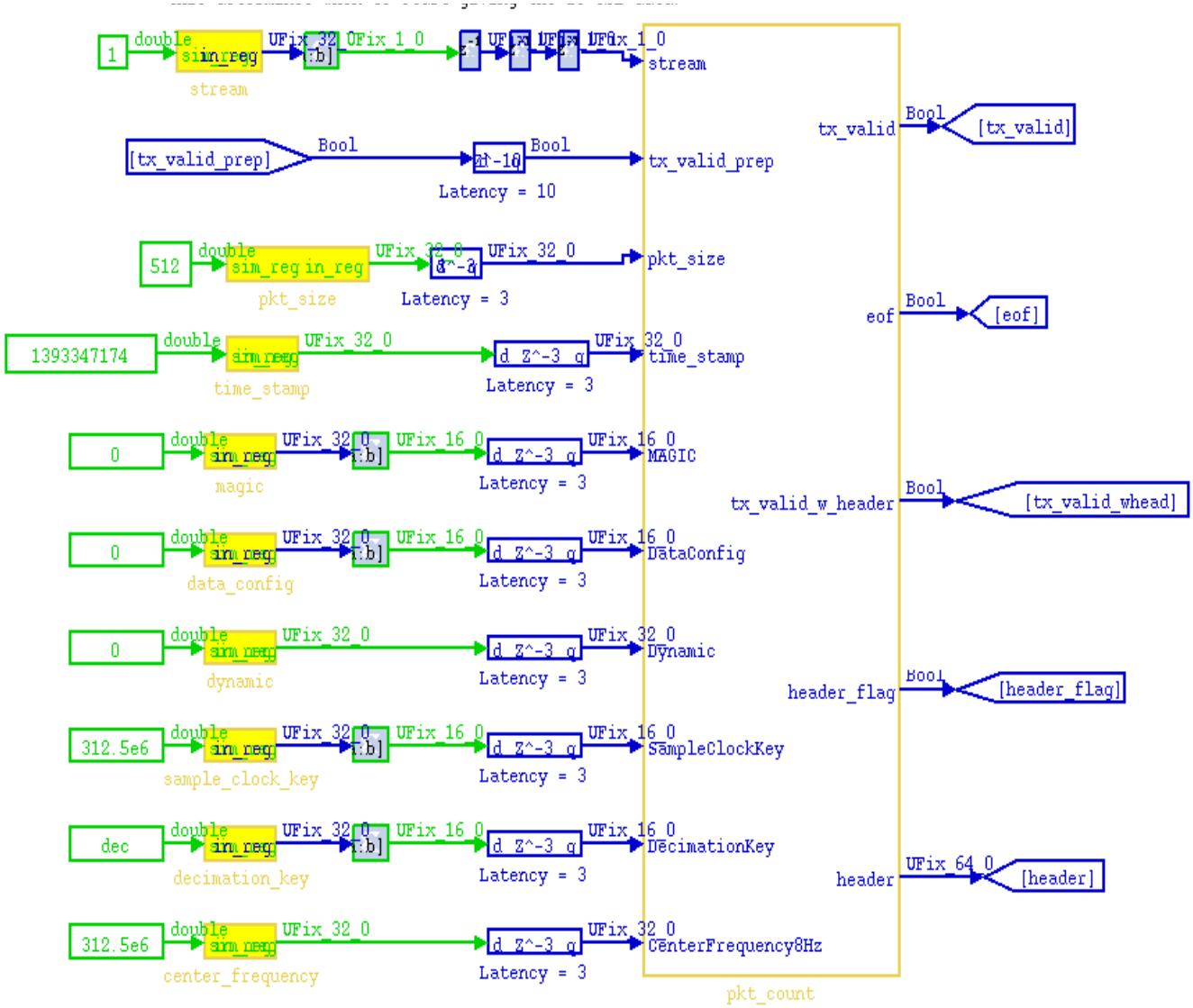


Figure 3.26. Control logic block ‘pkt_count’ for GbE yellow block.

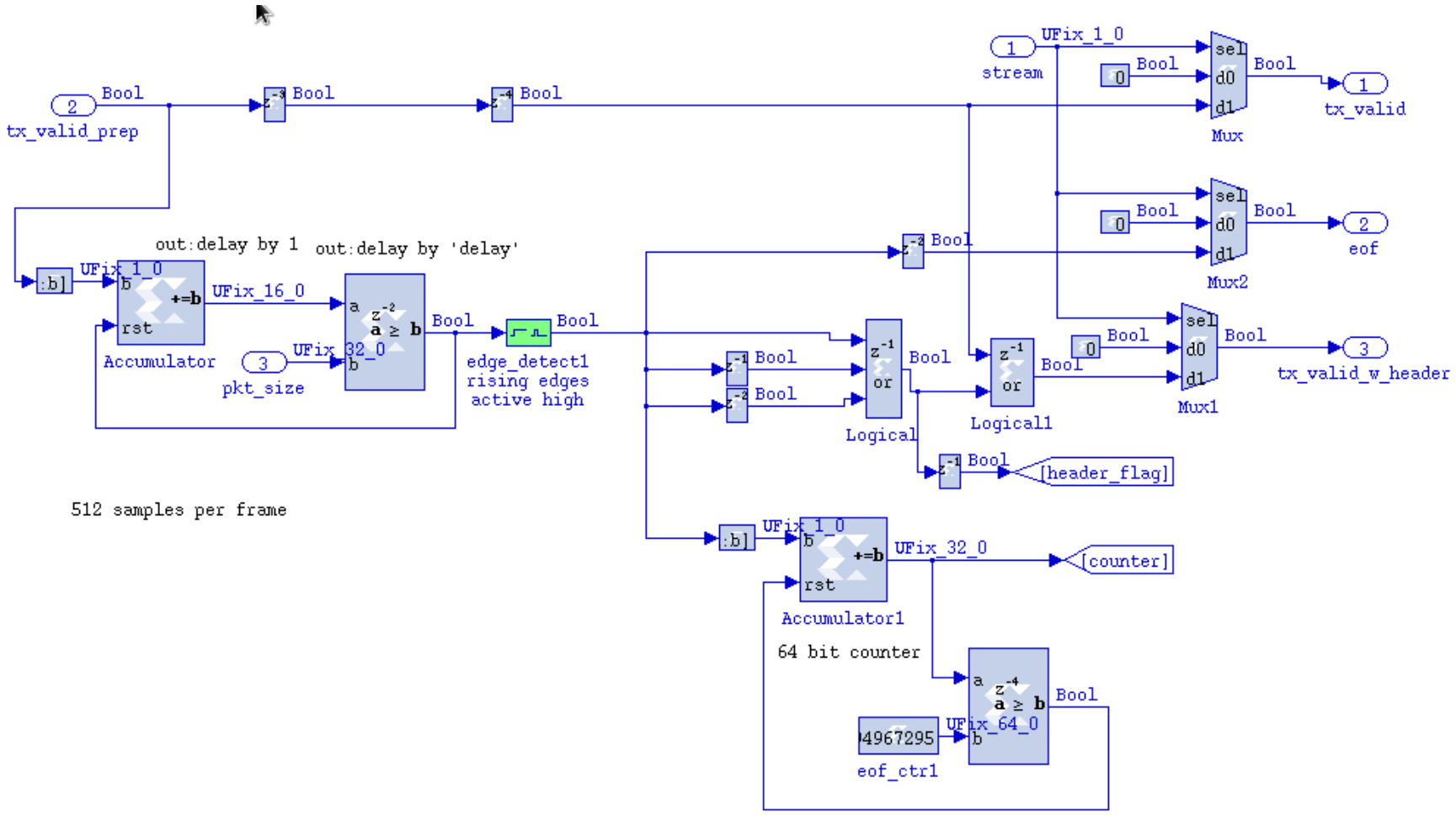


Figure 3.27. detailed control logic inside of ‘pkt_count’ subsystem.

3.6.3 *Header Insertion Strategy*

A 3-word-long header is inserted for each package. One purpose of the header is to help users characterize the data in the package, such as the sample rate, central frequency of the digital down converter, ADC clock rate, time stamp, data format, etc. Table 3.2 lists the header contents and their length. Furthermore, the header contains a frame counter 'FrameCtr' which is used to detect data loss during transmission. The header must be inserted when the 'eof' signal is high, since it represents the start of a new package. The header is inserted into the data stream through a Mux block, so the header must also be not inserted when signal 'DataRd' is high, but is inserted when the DataRd is low. This makes sure that the header won't replace the samples that will be written into GbE buffer.

Table 3.2. Header Information

Order of 64-bit words	**Name**	**Data type**	**Function**
First word	FrameCtr	unsigned 32 bits integer	Data frame counter. It's used to detect missing data as well as endianness of data
	Time	unsigned 32 bits integer	the conventional Unix time, precision to one second
Second word	MAGIC	unsigned 16 bits integer	something unchanging to look for in every header
	DataConfig	unsigned 16 bits integer	a key into a dictionary describing data configuration
	Dynamic	unsigned 32 bits integer	encode things that can change, such as gain settings
Third word	SampleClockKey	unsigned 16 bits integer	sample clock frequency key to dictionary
	DecimationKey	unsigned 16 bits integer	Decimation (bandwidth) control key to dictionary
	CenterFrequency8Hz	unsigned 32 bits integer	Center frequency of DDS, specified to 8 Hz

3.7 Discussion of the Arithmetic Operation Count

Data processing in the FPGA is performed in fixed point. Some control logic also involves significant math operations, such as the counters. In order to estimate the computational load, here we calculate the number of explicit multiply-accumulate (MAC) operations in the signal processing path only.

A typical design includes the blocks of a digital mixer (numerical oscillator, digital down converters), a polyphase filter with decimation, and a CIC filter. Table 3.3 lists the number of MAC operations (fixed point) at the FPGA clock (312.5MHz) per antenna, per channel for each block. The sum of MACs of these three blocks is 142.2, corresponding to 45 G MACs per second:

$$142.2 \text{ MACs}/clock\ cycle \times 312.5\ MHz \approx 45\ G\text{MAC}/sec \tag{3.33}$$

Table 3.3. MAC operations for the block of the mixer, polyphase filter and CIC filter per antenna per channel

Function	**Task**	**MAC count**	**Equivalent MACs** @312.5MHz	**Total**
Mixer	NCO phase increment	1 MAC @ 2500MHz	8	24
	sine/cosine is pre-calculated and stored in the look up table (LUT), so it doesn't involve runtime computing.		0	
	complex multiplication with incoming data	2 MACs @ 2500MHz	16	
Polyphase filter	8 sub-filters (each with 6 coefficients)	6 MACs x 8 x 2 (I&Q)	96	112
	Decimation(averaging)	8 MACs x 2 (I&Q)	16	
CIC filter (considering 6 stages and decimation R=30)	IIR(6-stage integration)	6	6	6.2
	FIR(6 stage comb filter)	6/R	0.2	
Total				142.2

We can fit up to 4 antenna inputs and 3 channels into the FPGA fabric. Therefore, the total number of MAC operations are 45G×4×3 = 540 G MAC per second for signal processing. This is

comparable to a 540-computer workstation cluster, assuming each computer has 1G MAC/s computation power.

3.8 FPGA LIMITATION: CLOCK RATE, RESOURCE USAGE

In practice, the FPGA design is limited by the maximum clock speed and available resources such as RAM slices, DSP slices, regular slices, and the auto-routing capability of the compiler. In general, the lower the FPGA clocks, the less the timing constrain is, and therefore, the easier the routing is, therefore many FPGA designs run at 200 MHz or less. However, for our receiver, in order to match the 2.5 GHz digitizer, the FPGA clock is 312.5 MHz, which approaches the maximum FPGA speed (350 MHz); after considerable effort we achieved routing solutions.

Another limitation of the FPGA design lies in the number of DSP block slices and memory (block RAM) slices. Some functions, such as a FIR filter, are achieved with DSP block slices. For example, a 48-tap FIR filter requires 48 DSP blocks. Each I/Q channel requires two filters, so 4 antenna inputs require 384 ($4 \cdot 2 \cdot 48$) DSP blocks and 3 frequency channels will further increase the number of filters and thus the use of DSP blocks by 3. The total number of available DSP slices is 2016. There will be difficulties if other functions use DSP blocks (e.g. multiplication) or more frequency channels are expected. And, as the FPGA resource usage gets larger, routing becomes more difficult due to less room and flexibility to map logic.

Table 3.4 lists the FPGA resource usage for 1-channel design and 2-channel design. Comparing the two designs, slices registers for 2-channel design are more than twice than the 1-channel design. As the design gets complex and more FPGA resources are used, the time-enclosure becomes more difficult in such high clock rate. Timing constraints are resolved by using pipelines which use additional slice registers. When a design uses more than half of the total slices registers, as for 3-channel design, even adding delay could not map the design onto the physical board. In this situation, we have to make a significant change on the design, such as using regular slice register for multiplication operation, instead of DSP481 block to relax the mapping effort, using fewer bits for the signal to release some FPGA resource for mapping.

Table 3.4. FPGA Resource Usage for 1-channel Design and 2-channel Design

FPGA Xilinx Virtex 6	Total number	1-channel design	2-channel Design
DSP48E1	2016	512 (25%)	1152 (57%)
Slice Registers (mainly used as FF)	595,200	51,343 (8%)	149,360 (25%)
Slice LUTs (mainly used as logic & memory)	297,600	16,819 (5%)	70,375 (23%)

The DSP48E1 blocks are mainly used by the polyphase filter. From Table 3.4, it also shows that in 2-channel design, the number doubles and is half of the total number of DSP blocks. We expect 4-channel design to be very difficult to fit in one Virtex 6 FPGA without some compromise, such as reducing the number of filter coefficients or the bit width.

3.9 System Performance Simulation

System performance parameters, such as SNR, are examined by simulation in Simulink. The simulation before test on board is necessary for two reasons. First, because the FPGA works with signals that are fixed point representations, quantization and truncation noise will be injected and accumulated during the DSP flow, affecting the SNR and noise level. Second, filtering and decimation operations increase processing gains in a predictable way; by comparing the simulation and theoretical results, the design can be verified.

We performed 4 simulations with different signal sources: single sine wave, double sine waves, single 6 MHz-wide linear chirp signal, and double 6 MHz-wide linear chirp signals, each with the same white noise. In the double signal test, the two signals have the same power, with one in pass band, and the other out of band.

The processing chain in the FPGA includes the digitizer, digital mixer, polyphase filter with decimation, and CIC filter. Figure 3.28 shows a simplified block diagram of signal propagation. The label of the signal is identical to label in Figure 3.4. It represents signal after each stage.

Figure 3.28. Block Diagram of the Processing Chain.

The signal source is the sum of one of above signals $a(t)$ and noise $n(t)$. ADC can only accept signals within [-1,1], so all signals are chosen to have an amplitude a_0 of 0.8. The noise amplitude has a standard deviation of 0.01. This is just big enough to trigger the least significant bit of the 8-bit digitizer. The sum of signal and noise is close to, but less than, the full range of ADC.

Table 3.5 shows the results of the SNR change during the processing chain. The sine wave power is calculated with $Ps = (a_0)^2/2$ from its amplitude a_0 and assuming characteristic impedance of 1 ohm. The power of Gaussian white noise with zero mean is calculated as $P_n = \sigma_n$, where σ_n is the variance of noise. During the processing chain, the power of the signal is measured by the variance of the signal, due to the fact that all the signals (and their sums) are zero mean. SNR is calculated with SNR = P_s/P_n, where the power of the noise P_n is measured independently by presenting only noise at the ADC input. For double signal mode, P_n is replaced with $P_{(n+s2)}$, the sum of noise and the other out-band signal.

Table 3.5. The Changes of SNR During Processing Chain

Simulation Input waveform	ADC In (A)	ADC Out (B)	Mixer Out (C)	PPF Out (D)
Single sine wave	P_s[5]= 0.32 watts $P_n = 10^{-4}$ watts SNR[6] = **35 dB**	P_s=var(sig)*=0.32 P_n=var(n) = 10^{-4} SNR = **35 dB**	P_s= 0.32 P_n= 10^{-4} SNR = **35 dB**	P_s= 29 P_n= 10^{-3} SNR =**45dB**
Double Sine waves	$P_{s1+s2+n}$ = 0.08 P_{s2+n} =0.08/2 =0.04 SNR[2] = **29dB**	$P_{s1+s2+n}$= 0.16 P_{s2+n}=0.08 SNR = **29dB**	$P_{s1+s2+n}$ =0.16 P_{s2+n}= 0.08 SNR = **29dB**	$P_{s1+s2+n}$ =7.26 P_{s2+n}= 0.0011 SNR =**38dB**
Single linear chirp signal	**34dB**[7]	**34dB**	**34dB**	**44dB**
Double linear chirp signals	**29dB**	**29dB**	**29dB**	**28dB**

[5] The signal and noise power is measured by the variance of the signals with/without signal present to the digitizer: P_s= Psig = var(signal) ; P_n = Pnoise = σ_n

[6] SNR = Ps/Pn for single input; SNR = ($P_{s1+s2+n}$- P_{s2+n})/P_n for double inputs

[7] Wideband sig calculation is the same as sine wave.

For the full rate data, the quantization noise is fairly high, but improves significantly as it passes through the FPGA signal processing. A single full scale bandlimited signal x_t, as shown in Figure 3.28, would have an SQNR of about 34 dB. The digital mixer does not degrade the SQNR because the precision of the sine and cosine (12 bits) significantly exceeds the precision of the sampled data. The polyphase Filter has a processing gain of 10 dB which improves the SQNR to 44 dB, for data d_t in Figure 3.4. A 6th order CIC filter has a processing gain of approximately 2.53×R where R is the decimation rate of the filter. For R = 25 (a typical choice, yielding 12.5 MHz output bandwidth) the CIC increases the SQNR by 18 dB, for an output SQNR of 62 dB, which would require at least 12-bit output samples to yield the full quality data.

The results indicate the design is robust, with little noise contributed by the digital processing. The SNR behaves similarly for sine waves and wideband signals; it behaves the same when there is interference at the stopband in polyphaser filter. The polyphase filter is able to suppresses the stopband signal without injecting noise or noticeable stopband residue to the passband.

Identical SNR results are achieved for different types of signals, showing that our design is robust and functional. Moreover, by carefully selecting signal bit width and scaling for each processing stage, it maintains signal quality without introducing additional truncation and overflow. Above is the analysis for a typical design. For our more complicated designs, the signal bit width is further optimized for allowing use of more FPGA resources in order to meet time closure, meeting time closure. For example, with 16-bit output precision, the signal can have at most 15x6=90dB dynamic range. We will explain later that 8 bit output precision would be sufficient to retain DTV signal quality in the real environment. By using 8-bit output instead of 16 bit, we are able to fit a third frequency channel in our design, which makes our design more powerful.

As we will also see later that subsequent coherent integration in floating point will further increase the dynamic range. The noise figure of the system is mainly limited by the precision (ENOB) of digitizer. For the passive radar application, however, the system is not noise limited because direct path and ground clutter are the main sources interference with target returns for a single receiver passive radar. The MRR uses a split receiver technology to deeply suppress the direct path and ground clutter [5]. We will address the issue of direct path and ground clutter in Chapter 4.

3.10 SIMULINK DESIGNS AND RESULTS

The designs vary in terms of different sampling rates, number of frequency channels, number of output bits. The designs with 2.5 GHz sampling rate support 4 RF inputs, and the designs with 5GHz sampling rate support 2 RF inputs. Table 3.6 lists the 6 typical FPGA designs. We will discuss each design and corresponding field test.

Table 3.6. FPGA algorithm designs

Design name	Sampling rate, number of frequency channel, number of output bit
Design 1 (2.5-1-8)	2.5Gsps, 1 frequency channel, 8-bit output
Design 2 (2.5-2-8)	2.5Gsps, 2 frequency channels, 8-bit output
Design 3 (2.5-1-16)	2.5Gsps, 1 frequency channel, 16-bit output
Design 4 (2.5-2-16)	2.5Gsps, 2 frequency channels, 16-bit output
Design 5 (5-1-8)	5Gsps, 1 frequency channel, 8-bit output
Design 6 (2.5-3-8)	2.5Gsps, 3 frequency channels, 8-bit output

3.10.1 *First Design: 2.5 GSPS Mode, 1 Frequency Channel, 8-bit Output*

The first design supports 4 antenna RF input with ADCs sampling at 2.5GHz and one frequency selection. The available output bandwidth is 78.125 MHz maximally, and output data is 8-bit wide for both I and Q. Figure 3.29 shows the spectrum of the KONG DTV signal (centered at 575 MHz and 6 MHz wide) of the 4 antennas with Matlab Welch method. During run time, the central frequency is tuned to 575 MHz, and CIC filter decimation is set to 30, corresponding to about 10 MHz bandwidth. 6 bits out of 8 bits are used to represent the signal fed into the ADC, and data width of the output is 8 bits. Over 50 dB dynamic range is observed from the spectrum. It explains that the signal processing, especially the two filtering operations, adds 2 more useful bits to the signal of interest. The non-flat spectrum for the DTV signals for antenna A, B and D are caused by frequency-selective fading in the urban propagation environment from other buildings and trees. Also the 25-meter-long coaxial cable between the antenna and sampler introduces its own frequency selectivity because of sharp bends.

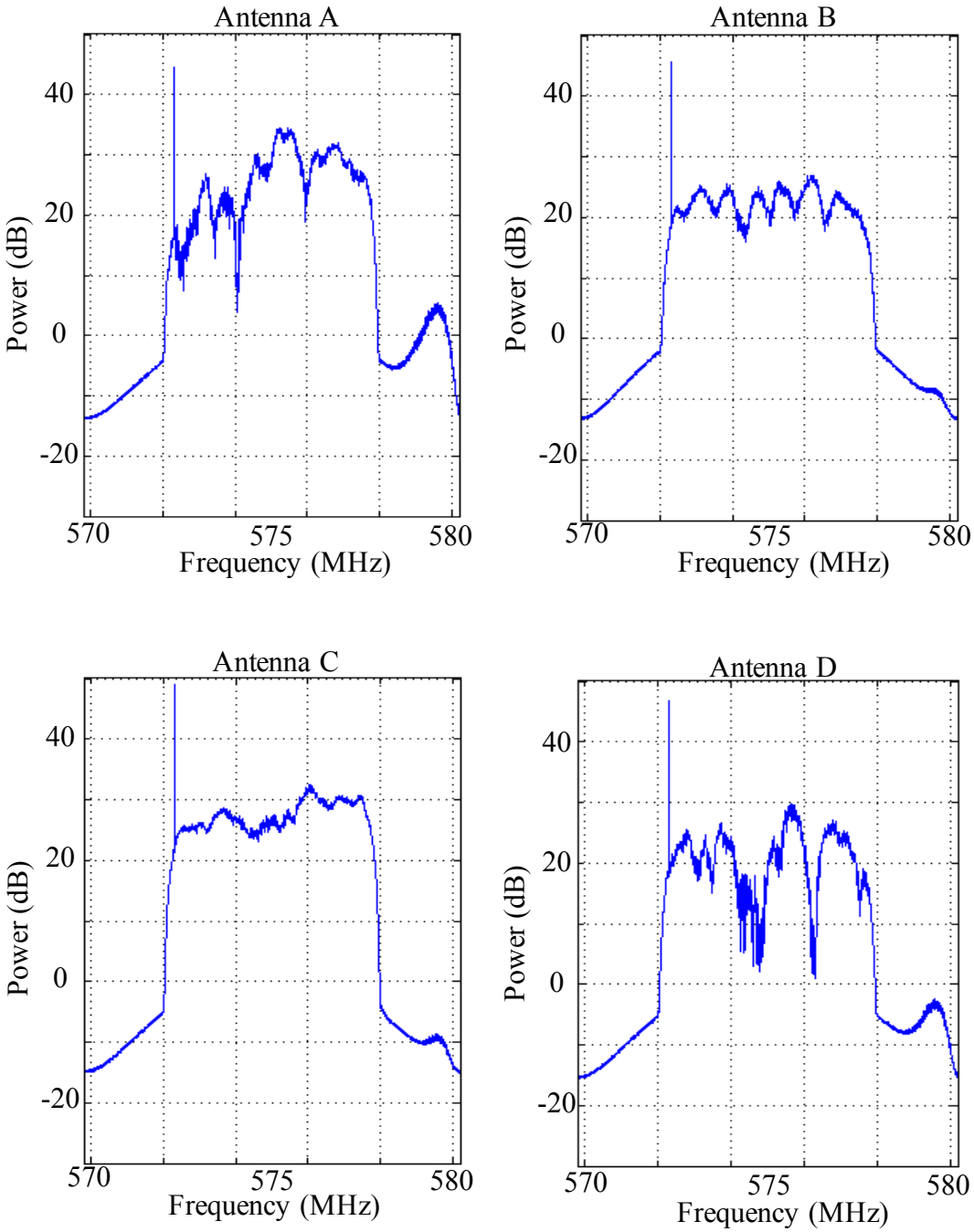


Figure 3.29. KONG DTV channel captured with the first FPGA design (2.5 GHz, 4 antennas, 1 frequency, fc=575MHz, bandwidth=10.4167MHz)

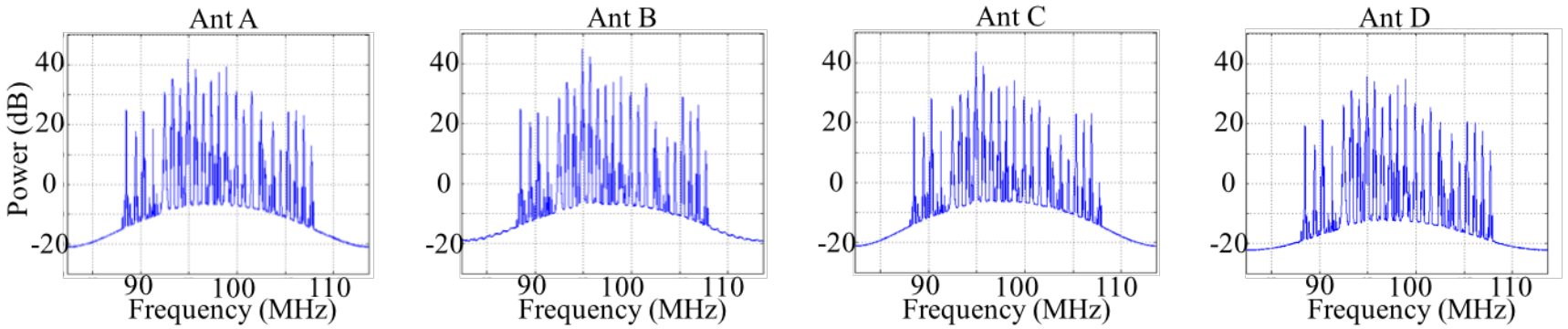
3.10.2 *The Second Design: 2.5 GSPS Mode, 2 Frequency Channels, 8-bit Output.*

The second design is similar to the first design, but adds a second channel. The data of the second channel are passed through a separate 10 GbE link. With the second channel, user can select 2 frequency channels simultaneously during run time. This design required extensive effort to achieve timing closure. Not only does compiling a single design take about 5 hours, but several attempts are generally needed in order to close timing. Therefore, it could take several weeks to complete a design for just trouble-shooting the timing. Any architecture change or design mistake would cause a new development cycle. Table 3.7 compares the FPGA resource usages of the first and second design. It shows that 2-frequency mode uses about twice that of the 1-frequency mode. This makes sense because the block that adopts DSP slices are those DDS compiler, FIR compiler, CIC compiler blocks which doubled from 1-frequency mode to 2-frequency mode. However, the slice registers used are more than twice as the 1-frequency mode. This is due to the fact that more registers were used for routing to meet time disclosure.

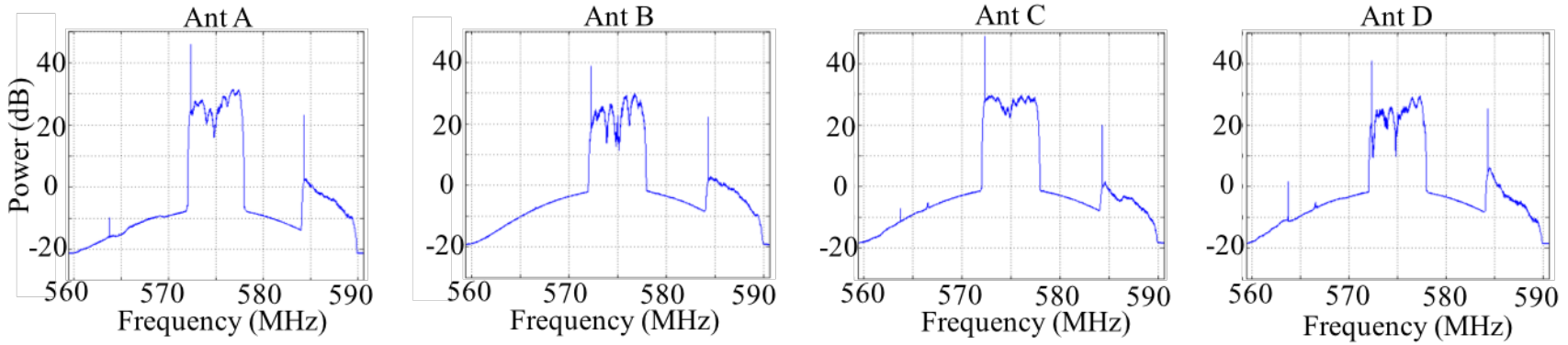
Figure 3.30 shows the spectra of the KONG DTV signal and the entire FM broadcast band signals which present at the antenna at the same time. The parameters for the first frequency branch include central frequency f_0 of 98MHz and CIC decimation factor R_0 of 10; the parameters for the second frequency branch include central frequency f_1 of 575 MHz and CIC decimation factor R_1 of 10. The top shows the channel of entire FM spectrum for 4 antennas and the bottom shows a KONG Digital TV station at 575MHz and an adjacent station at 587 MHz. The details of the spectrum on the left of Figure 3.30(c) show a few adjacent FM stations, and spectrum on the right shows the KONG DTV station with 6MHz bandwidth. The fluctuation in the DTV spectrum is due to frequency selective fading and coaxial cables as mentioned above.

Table 3.7. FPGA resource comparison of first and second design. First design implements one frequency channel and the second design implements two frequency channels. The second design uses more than twice FPGA resources, which is due to the fact than extra registers are sued as pipelines to meet timing.

	Total number	First Design	Second design
DSP48E1 (mainly used for DSP)	2016	512 (25%)	1152 (57%)
Slice Registers (mainly used as FF)	595,200	51,343 (8%)	149,360 (25%)
Slice LUTs (mainly used as logic, memory)	297,600	16,819 (5%)	70,375 (23%)



(a) Channel 1 mixer frequency f1=98MHz, bandwidth = 31.25MHz; vertical divisions are 10 dB



(b) Channel 2 mixer frequency f2=575MHz, bandwidth = 31.25MHz; vertical divisions are 10 dB

Figure 3.30. Spectrum of the data taken in the 2.5 GHz mode (4 RF inputs, sampler rate at 2.5 GHz) with 2 channels (mixer frequencies at f_1 = 96 MHz and f_2 = 575 MHz, output bandwidth of around 30 MHz for both channels). (a) Spectra of channel 1 (FM channel); (b) Spectra of Channel 2 (DTV channel)

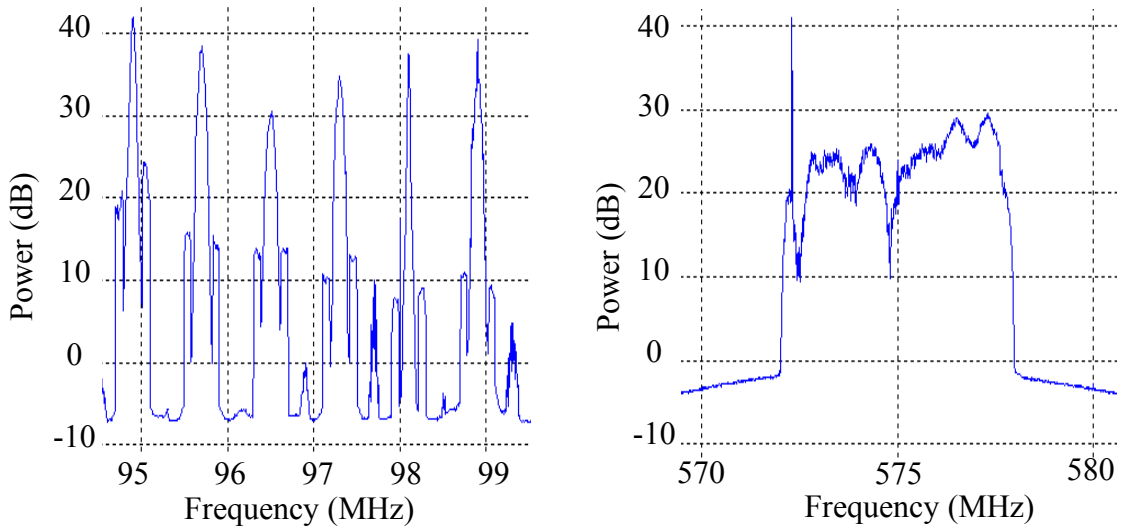


Figure 3.31. Detailed view of Antenna D spectra in the two channels in Figure 3.30. Note that the digital side bands are clearly visible for the FM stations.

3.10.3 *The Third Design: 2.5 GSPS Mode, 1 Frequency Channel, 16-bit Output.*

The third design is similar to the first design, but outputs 16-bit data, instead of 8-bit data. The 16-bit width is useful when a strong signal in the environment dominates the total power present at the ADC input. The 8-bit width has a maximum dynamic range 42 dB assuming 6 dB/bit, while 16-bit width has a maximum dynamic range up to 90 dB. It avoids saturation when the signal to truncation noise ratio is larger than 40 dB. Figure 3.32 shows the spectrum of KONG DTV signals captured with the third design. The SNR is identical to the SNR in first design, indicating that the 8-bit output is sufficient to retain the original signal to ADC truncation noise ratio for KONG DTV signal.

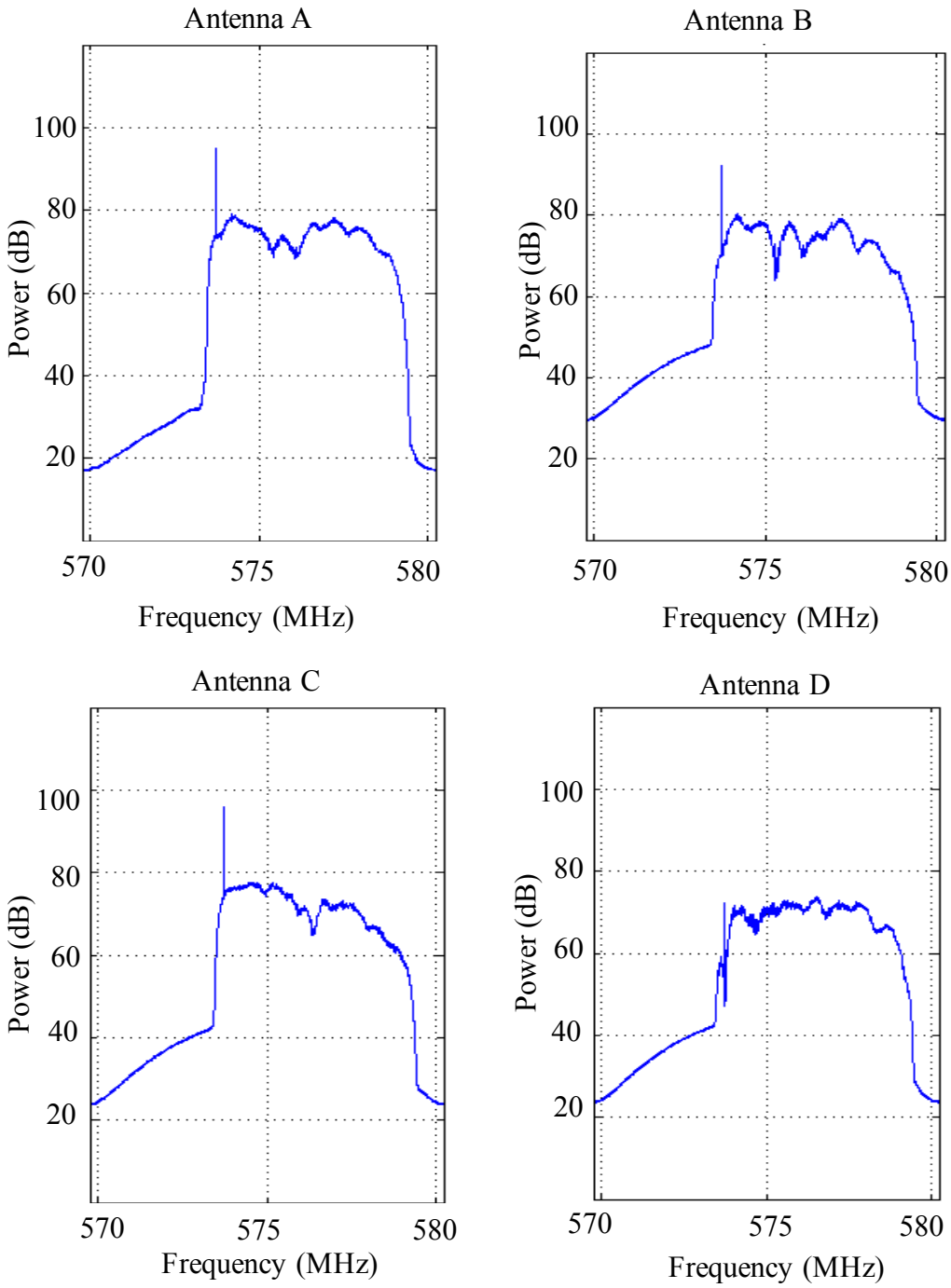


Figure 3.32. Spectrum from the second design (2.5GHz, 4 antenna, 16 bit output, fc=575MHz, bandwidth = 10.4167MHz).

3.10.4 *The Fourth Design: 2.5 GSPS Mode, 2 Frequency Channels, 16-bit Output in One Data Stream.*

The fourth design is similar to the third design (1 frequency channel, 16-bit output), but adds a second frequency channel. Different from the second design (2 frequency channels, 8-bit output), before the data is sent out, the two data streams for the 2 channels are added together to become one stream. Compare with the two channel with each data stream 8 bit wide, 1 stream in 16-bit width improves the signal to noise ratio. However, the timing failed after several attempts to relax time constrains and close the critical path (signal propagation in FPGA). The reason for timing failure is due to the large register consumption from the auto-bit selection module and doubled DSP consumption of the second channel. As more FPGA slices are used, automatic mapping during compilation doesn't have enough flexibility to optimally distribute the logic so that every signal propagation time from one slice to another meets the minimum timing requirement.

3.10.5 *The Fifth Design: 5 GSPS Mode, 1 Frequency Channel, 8-bit Output*

The fifth design is similar to the first design, except that the ADC samples at 5 GSPS. One stream RF signal at the ADC input is digitized into 16 parallel data streams. The passband of the polyphase filter is 1/16. In normalized form, instead of 1/8 in the first design. Figure 3.33 shows the entire spectrum with a short shot of data from ADC and with antenna located on the roof of a 4-floor building. The spectrum covers from DC up to 2.5 GHz without aliasing. You can find FM broadcast at around 0.1GHz, TV broadcast between 200MHz to around 800MHz, cellular activities sparsely located at higher frequency. WiFi signal at 2.4 GHz can be detected when antenna is put inside of the building. GPS signals at 1575.42 MHz (L1 band) and 1227.60 MHz (L2 band) can be found with appropriate demodulation/decoding. The peak at 1.25GHz is an artificial signal. It's generated due to the imperfect calibration of the four ADC cores, which are sequentially multiplexed thus introducing a 1.25 GHz signal. This mode enables the receiver to capture any signal in this [0, 2.5 GHz] frequency range. As an example, Figure 3.34 shows the cellular spectrum centered at 1.955 GHz. It would be aliased in the 2.5 GSPS mode.

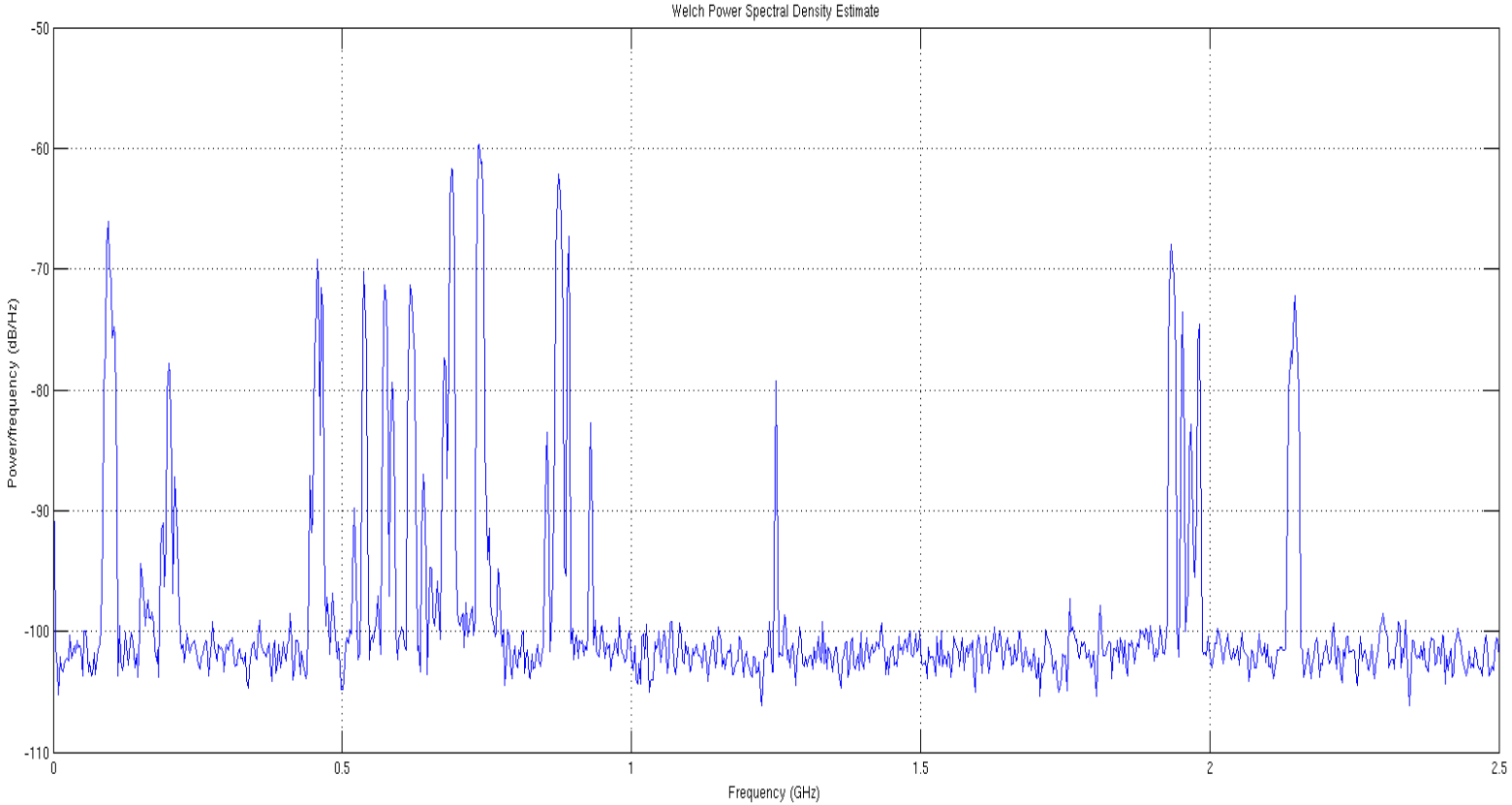


Figure 3.33. Spectrum of raw ADC sampled data with snapshot.

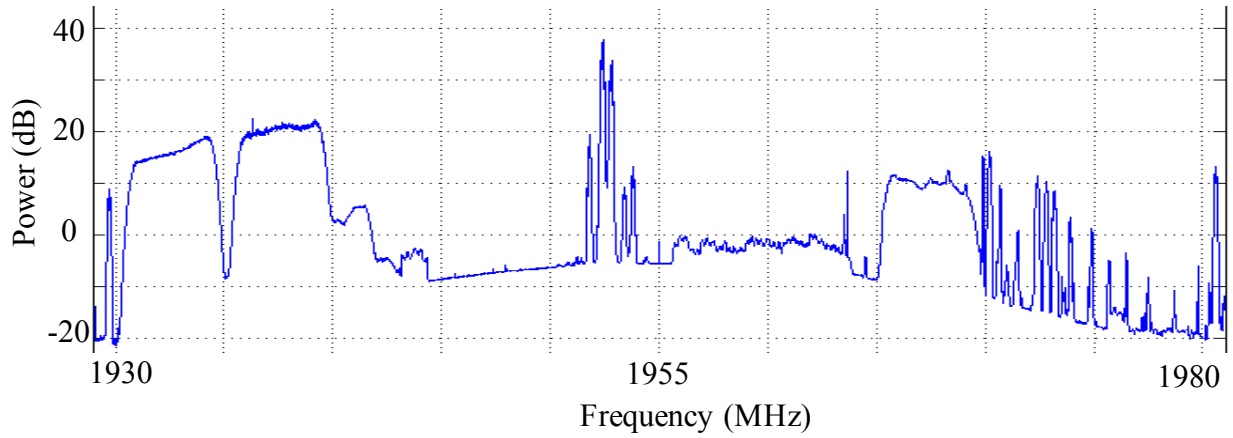


Figure 3.34. Spectrum from the fifth design.

3.10.6 *The Sixth Design: 2.5 GSPS Mode, 3 Frequency Channels, 8-bit Output*

The sixth design is similar to the first design, but has 3 fixed frequency channels, two of them with bandwidth 312.5 MHz, and the other with bandwidth 10MHz. Obviously, the 312.5 MHz bandwidth channels eliminate the CIC filters in the processing flow. The purpose of the design is to make full use of the 8 10GbE ports on the board and output bandwidth as wide as possible. Because the current program for 10GbE transmission could only support up to 6 Gbps, which corresponds to about 120 MHz bandwidth for 4-antenna, 8-bit IQ data, the eight 10GbE ports can achieve 800MHz bandwidth in total. Table 3.8 compares the FPGA resource usages between the first and the sixth design.

Table 3.8. FPGA Resource Comparison of First, Second and Sixth Design

	Total	**First Design** (2.5 GSPS mode, 1 channel, 8-bit output)	**Second design** (2.5 GSPS mode, 2 channels, 8-bit output)	**Sixth design** (2.5 GSPS mode, 3 channels, 8-bit output)
DSP48E1 (mainly used for DSP)	2016	512 (25%)	1152 (57%)	1,592 (78%)
Slice Registers (mainly used as FF)	595,200	51,343 (8%)	149,360 (25%)	169,697 (28%)
Slice LUTs (mainly used as logic, memory)	297,600	16,819 (5%)	70,375 (23%)	87,630 (29%)

3.11 AMBIGUITY FUNCTION ANALYSIS

In order to test the receiver system, we recorded a DTV station (KONG) centered at 575 MHz with bandwidth 6.25 MHz and estimated its self-ambiguity. The (Woodward) ambiguity function describes the ability of a radar waveform to resolve targets in range and Doppler [81].

Figure 3.35 contains a plot of the power spectrum of KONG received on two antennas (Antenna A and Antenna B) with data taken on a pair of antennas in the 5 GHz mode at an output sample rate of 6.25 MHz. The output format of data is 16 bits in this case. In the United States, Digital TV broadcasts use 8VSB modulation, which has a pilot tone at the low frequency edge easily visible in the spectra. The distorted, non-flat spectrum indicates frequency selective fading caused by multiple reflections along the propagation path, highlighting the antenna position influence on the observed signal.

Figure 3.36 shows the amplitude distribution of the data (real part of the complex-valued data). As the 16 bit is interpreted as signed integer, the maximum magnitude is 2^{15}. It is Gaussian distributed with zero mean and standard derivation of $7x10^3$. The maximum threshold of the average output value is set at $2^N/4$, where N is the number of bit at the output. The choice of threshold is based on the optimization of the truncation and quantization noise, which needs to be considered for the FGPA and its fixed point computations.

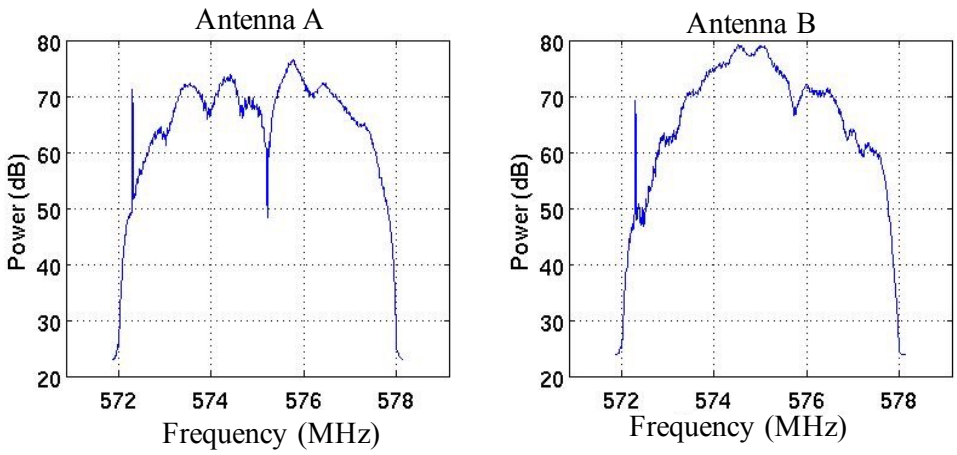


Figure 3.35. Power spectrum of the two antennas of data centered at DTV station KONG (575MHz) with bandwidth 6.25 MHz. It has a pilot tone at the low frequency edge easily visible in the spectra. The distorted, non-flat spectrum indicates frequency selective fading caused by multiple reflections along the propagation path including within the coaxial cable between the antenna and digitizers, highlighting the antenna position influence on the observed signal.

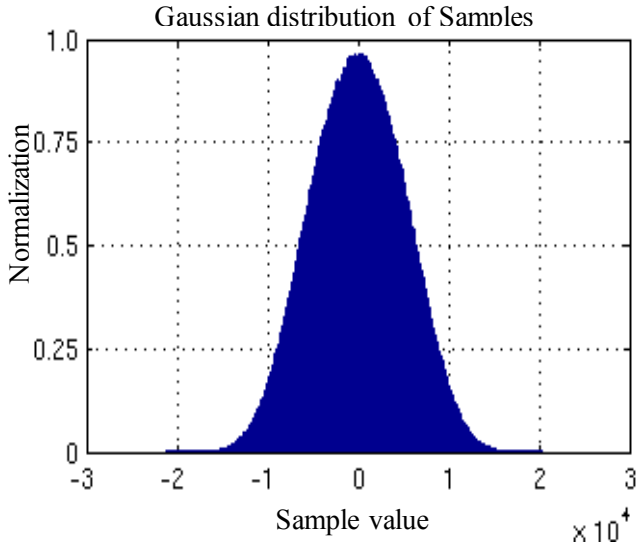


Figure 3.36. Histogram of amplitude distribution of the data (real part), showing the expected normal distribution. The sample is 16 bit wide, with a theoretical maximum magnitude of 2^{15}. The auto-selection module sets the maximum threshold at a quarter of 2^{15}.

Figure 3.37 (a) shows the self-ambiguity with measured from 1 second of data. The coherent integration time is 42 mili-seconds, in which 256-point power spectra are estimated with Doppler velocity range +/- 825 m/s. Then 24 of these spectra are incoherently averaged to form the cross ambiguity estimate. This is an appropriate spectrum estimation algorithm for the study of ionospheric turbulence. The intrinsic range resolution for DTV is inversely proportional to its bandwidth of 5.38 MHz; in backscatter equivalent this corresponds to 28 m. From the coherent integration time and the DTV bandwidth, we estimate a processing gain of 5.38 MHz × 42 mili seconds = 2.3×10^5, or 53.5 dB. This closely matches the signal to clutter ratio of the range-Doppler ambiguity in Figure 3.37(b). Here, we didn't attempt to suppress the transmitter ground clutter. We have shown that, in our paper [38], DTV signals captured by this receiver are successfully demodulated; clutter floor is further dropped with adaptive filtering technique to provide over 100 dB dynamic range for detecting week targets.

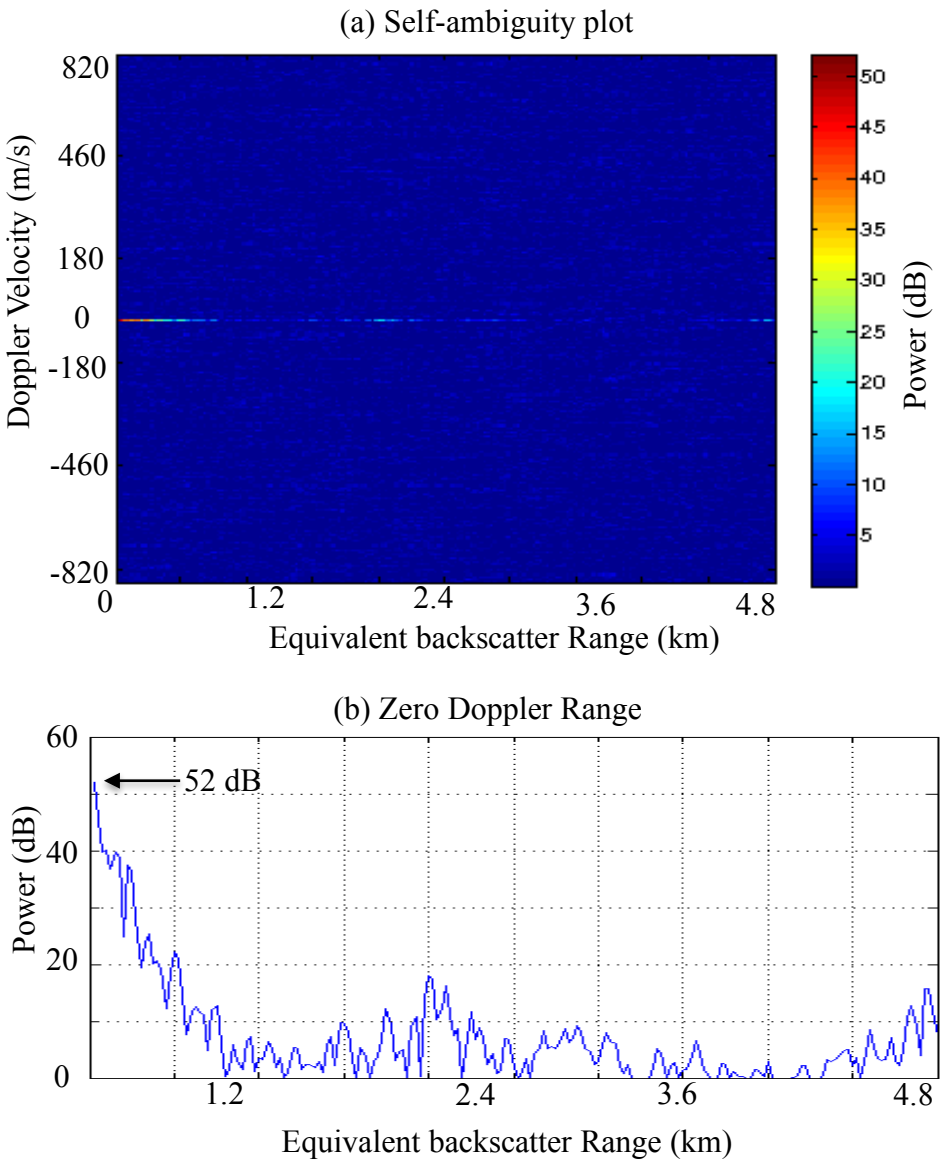


Figure 3.37. Self-ambiguity with 1 second of data. (a) shows the ambiguity function in range and Doppler velocity; (b) plots Zero Doppler vs. Range of Ambiguity function in (a).

3.12 SUMMARY

In this chapter we presented a thorough analysis of the FPGA designs. Each stage of the signal flow is introduced, from the principle to the related Xilinx blocks and control logic design. The ADC is also tested and calibrated with a small amount of data captured in FPGA (snapshot). In

this chapter, we also demonstrated several designs from the run-time captured data. We compared the FPGA resource for different frequency channels and different ADC modes. Finally, the self-ambiguity of the captured DTV signal is computed with 1 second of data. 52 dB dynamic range is demonstrated. Direct path and several ground clutter signals can be clearly seen in the ambiguity plot.

The design is complicated, but as a result, it is very agile in the following aspects:

(1) The high speed samplers provide one tremendous advantage: greatly relaxed requirements on analog filtering. For the 5 GSPS samplers all signal power below 2.5 GHz is Nyquist sampled, and nearly any desired filter function can be realized in software. To put it a little differently, every set of high performance analog filters costs about 20% of the cost of the rest of the data acquisition system, and would require manual exchange or additional RF switch fabric. By sampling sufficiently quickly the receiver complexity is greatly reduced and the flexibility retained.

(2) It is easy to have a different signal processing configuration by reprograming the FPGA without modifying the hardware components, which makes it versatile for many different applications. For example, it can be configured to calculate ambiguity function by feeding data back into the FPGA. It can be configured for one RF input to perform more complicated processing and algorithm, or can be configured for multiple RF inputs (phased array) to achieve angle of arrival measurement. It can be configured to a total different processing flow other than the current design.

With a single RF input, the new receiver is capable of simultaneously extracting multiband channels from the fast sampler and FPGA processing ability. In the ISIS, system acquiring the DTV and FM channels at the same time requires duplicating the RF input signal path, one path with a mixer for DTV channel. In ROACH-2, there are twice as many DSP slices (which are specially designed for fast multiplier computation) as in ROACH-1, which makes single RF multiband channel possible.

(3) The down-conversion is performed numerically in the FPGA, instead of an analog mixer. Therefore, the two 90 degree shifted oscillators can be designed identically, unlike the analog mixer in which engineer has to deal with synchronization. Also one ADC is required instead of two since the down conversion is performed after the signal is sampled. This insures identical performance in the I and Q channels.

(4) The receiver is relatively easy for user to operate. People can operate it to acquire data they need without the knowledge of the FPGA programing and receiver hardware. Another feature in my design for ROACH2 is that the mixer frequency can be adjusted during run time. It avoids new design for new frequency requirement and also avoids halting the receiver when you switch frequency band.

Chapter 4. DETECTION

In previous chapters, we introduced the new digital receiver developed at the UW Radar Remote Sensing Lab is capable of directly sampling each of four antennas at 5 GSPS. With this receiver we are able to capture all signals below 2.5 GHz in support of our passive radar efforts in the VHF and UHF radio spectrum. For passive radar applications, the signal to clutter ratio is the usual performance limit for high power transmitters, such as DTV. Therefore, we must separate the strong direct path, multipath and the weak echoes of the object of interest from the total signals.

DTV signals have a known structure which permits recovery of the original transmitted waveform from imperfect reception [82]. With such a nearly ideal reference signal, we are able to map the multipath and also improve the instantaneous dynamic range to over 100 dB with a one-second coherent processing, and thus detect the weak echoes from targets of interest, such as aircraft or ionospheric field-aligned irregularities, and pave the way for AoA estimates or interferometric imaging of these scatterers.

We will explain the digital process for DTV demodulation. The performance of the digital process is examined by the estimation of error rate of the 8-level symbols [82, 83]. Finally, we will compare the numerical noise floor of the ambiguity function and thus the target detectability before and after the direct and multipath are removed from the total signals.

4.1 Adaptive Filter Algorithm

Adaptive filters constitute an important part of statistical signal processing. They offer a highly attractive solution to the problem of signals with unknown statistics, or signals in a statistically nonstationary environment where the signal propagation path is changing or even blocked. Our received DTV signals are non-stationary, and are propagated through an unknown environment channel with severe multipath. Adaptive filter theories and algorithms thus have been explored and have shown a significant improvement in performance over the use of a fixed filter designed by conventional methods. The most common filters include Kalman filter, LMS filter, RLS filter, etc. [60] We compared the performance of several filters on our signal. Based on the convergence rate, adaptive capability, computational complexity, we chose the appropriate filters for the DTV demodulation process in this chapter. Particularly, The Wiener filter is selected to initialize a filter based on the short DTV sync frame, due to the simplicity of the filter and stationary-environment

approximation in a short period of time. The LMS filter is selected to recover the transmitted waveform from the multi-path and noisy signal. The selection is made based on its simple implementation, low computation complexity, good convergence and excellent adaptive performance for the non-stationary environment. We will introduce these two filters below, and show the application of these filters during DTV demodulation and their performance.

4.1.1 *Wiener Filter*

The Wiener filter [60] defines the optimum linear filter for a stationary environment. Figure 4.1 shows the block diagram of the statistic filtering problem.

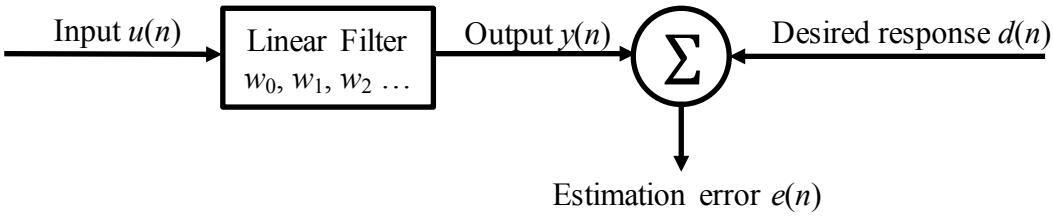


Figure 4.1. Block diagram of the statistic filtering problem.

The filter input consists of a time series $u(0)$, $u(1)$, $u(2)$, … and the filter is itself characterized by the impulse response w_0, w_1, w_2, … At some discrete time n, the filter produces an output demoted by y(n). This output is used to provide an estimate of a desired response designated by d(n). The Wiener-Hopf equation gives the filter w such that the mean square error of the difference of the output y(n) and desired response d(n) is minimal. Let R denote the M-by-M correlation matrix of the tap inputs u(n), u(n-1), …, u(n-M+1) in the transversal filter of Figure 4.1; that is,

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] \tag{4.34}$$

where

$$\mathbf{u}(n) = [u(n), u(n-1), \ldots, u(n-M+1)\ ]^T \tag{4.35}$$

is the M-by-1 tap-input vector. In expanded form, we have

$$\mathbf{R} = \begin{bmatrix} r(0) & \cdots & r(M-1) \\ \vdots & \ddots & \vdots \\ r^*(M-1) & \cdots & r(0) \end{bmatrix} \tag{4.36}$$

where r(n) is the the self-correlation at lag n. Correspondingly, let p denote the M-by-1 cross-correlation vector between the tap inputs of the filter and the desired response d(n):

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)] \tag{4.37}$$

In expanded form, we have

$$\mathbf{p} = [p(0), p(-1), \cdots, p(1-M)]^T \tag{4.38}$$

We may thus write the Wiener-Hopf equations in the compact matrix form

$$\mathbf{R}\mathbf{w_0} = \mathbf{p} \tag{4.39}$$

where w0 denotes the M-by-1 optimum tap-weight vector of the transversal filter (optimum in the mean-square-error sense); that is

$$\mathbf{w_0} = \left[w_{o,0}, w_{o,1}, \cdots, w_{o,M-1}\right]^T \tag{4.40}$$

To solve the Wiener-Hopf equations for w0, we assume that the correlation matrix R is nonsingular. We may then premultiply both sides of Equation by R^{-1}, the inverse of the correlation vector, obtaining

$$\mathbf{w_0} = \mathbf{R^{-1}p} \tag{4.41}$$

The computation of the optimum tap-weight vector w0 requires knowledge of two quantities: (1) the correlation matrix R of the tap-weight vector u(n) and (2) the cross-correlation vector p between the tap-input vector u(n) and the desired response d(n).

4.1.2 *LMS Filter*

Now we will apply adaptive filter algorithms for DTV demodulation. One of the algorithms is Least Mean Square (LMS) [84, 85]. Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that produce the least mean square of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on current error estimation. A significant feature of the LMS algorithm is its simplicity. Moreover, it doesn't require measurements of the pertinent correlation functions, nor does it require matrix inversion.

The block diagram of LMS algorithm is illustrated in Figure 4.2. The transversal filter is responsible for performing the filtering process, and the adaptive weight-control mechanism performs the adaptive control process on the tap weights of the transversal filter.

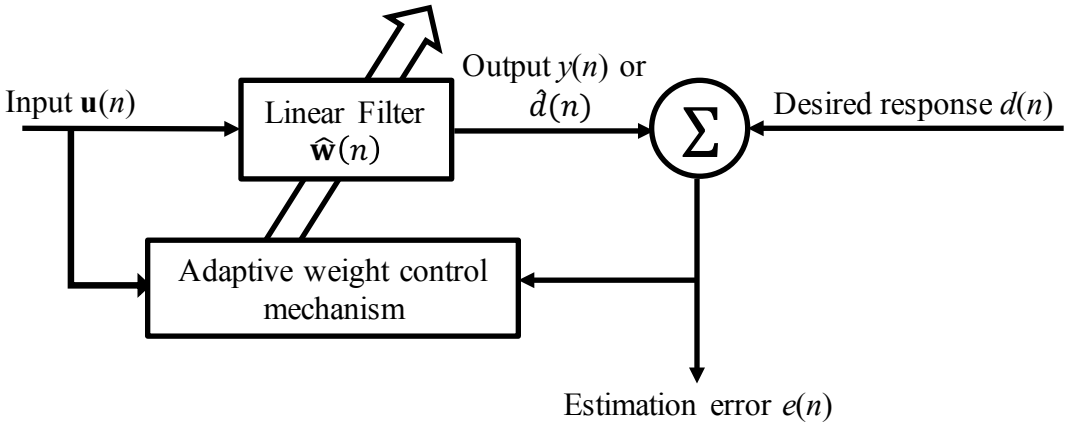


Figure 4.2. Block diagram of LMS algorithm.

w(n) is the transversal filter, represented as M-by-1 tap-weight vector, where M is the filter order. **u**(n) is a M-by-1 tap input vector. During the filtering process, the desired response $d(n)$ is supplied for processing alongside the tap-input vector **u**(n). Given the input, the transversal filter produces an output d'(n) or y(n) used as an estimate of the desired response d(n). Accordingly, we may define an estimation error e(n) as the difference between the desired response and the actual filter output, as indicated in the Figure 4.2. The estimation error e(n) and the tap-input vector u(n) are applied to the control mechanism, and the feedback loop around the tap weight is thereby closed. Equivalently, we may write the result in the form of three basic relations as follows:

1. Filter output:

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \tag{4.42}$$

2. Estimation error or error signal:

$$e(n) = d(n) - y(n) \tag{4.43}$$

3: Tap-weight adaptation:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n) \tag{4.44}$$

where μ is an adaptive factor, which adjusts the convergence rate of the the filter.

4.2 Recovery of Transmitted DTV Waveform

4.2.1 *DTV Waveforms*

Compared to FM and other commercial broadcast signals, Digital TV signals (DTV) have high average power, and larger bandwidths. It has elegant synchronization segments embedded in the signals which can be used to fully demodulate and recover the transmitted waveform as long as the SNR is sufficiently high. With the demodulated signal, we can achieve the same sensitivity as an active bistatic radar, while retaining all the merits of a passive radar, such as the inherent stealth, reduced power consumption, no need for transmitter license and low EM exposure.

Several regions of the world use different digital television broadcasting standards (DTB). The Digital Video Broadcasting - Terrestrial (DVB-T) adopted in Europe, Australia and New Zealand, uses coded orthogonal frequency-division multiplexing (OFDM) modulation. It has been widely studied for passive radar application [6] [8, 9, 27, 33, 35, 86, 87]. In North America, the Advanced Television System Committee (ATSC) uses eight-level vestigial sideband (8VSB) standard for terrestrial broadcasting [82], and it has significant structural differences from the DVB-T standard used elsewhere.

8-VSB is essentially an 8-level amplitude-shift-keyed (ASK) signal that has been Nyquist-filtered to a single sideband. it has 6MHz bandwidth and noise-like properties. DTV payload data

at 19.39 Mbps are augmented with error correction and overhead to 32.29 Mbps. These are used to modulate the 8- level (3-bit) symbols at a symbol rate of 10.76 MHz. Nyquist filtering using a root-raised-cosine filter having a roll off factor of alpha = 0.1152 passes the upper sideband (5.38 MHz) and limits the total bandwidth to the channel size of 6 MHz. Randomization and interleaving processes result in data that have correlation properties similar to a pseudo random sequence. The carrier is generated by adding a constant value of 1.25 to the symbol values of +/-1, +/-3, +/-5, +/-7. This accounts for about 7% of the signal power and is removed in mismatched filtering. Side-lobe features in the ambiguity function correspond to two framing codes. The first is the segment sync, a [+5, -5, -5, +5] sequence repeating every 77.3 μs (828 symbols). The longer field sync occurs at 24.2 ms intervals.

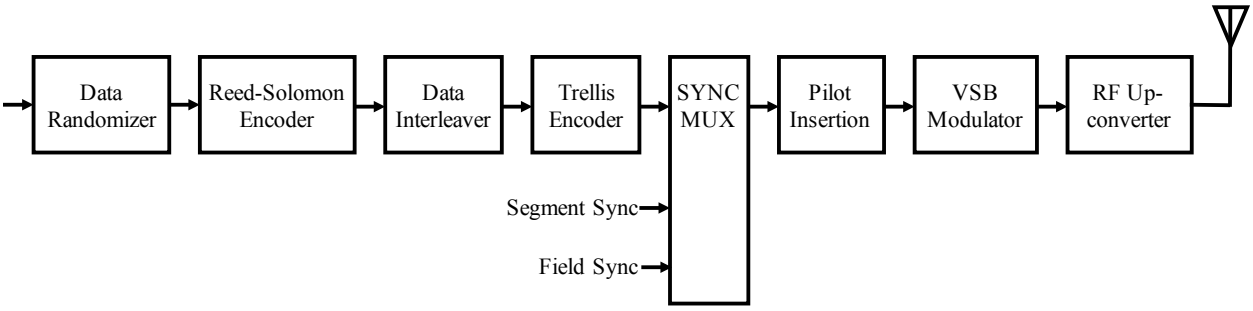


Figure 4.3. Block diagram of DTV modulation [82].

Figure 4.3 shows block diagram of DTV modulation. During DTV modulation, the bit stream of the coded video information is firstly channel coded through Data randomizer, Reed-Solomon encoder, Data Interleaver and Trellis encoder. In the data randomizer, each byte value is changed according to known pattern of pseudo-random number generation. This will enable the transmitted signal frequency response have a flat noise-like spectrum in order to use the allocated channel space with maximum efficiency. Reed Solomon encoding is a Forward Error Correction (FEC) scheme applied to the incoming data stream. It takes all 187 bytes of an incoming MPEG-II data packet to create 20 additional bytes (known as Reed-Solomon parity bytes) which are then tacked onto the tail end of the original 187-byte packet. It can correct up to 10 byte errors per packet. The data interleaver scrambles the sequential order of the data stream and disperses the MPEG-II packet data throughout time (over a range of about 4.5 millisecond through the use of memory buffers) in order to minimize the transmitted signal's sensitivity to burst type interference. Trellis coding [88] is another form of forward error correction – using convolutional codes. It is an evolving code that tracks the progressing stream of bits as it develops through time. In the trellis

coder, each 2-bit word that arrives is compared to the previous 2-bit words. A 3-bit binary code is mathematically generated to describe the transition from the previous 2-bit word to the current one. Therefore, the trellis coder in the 8VSB system is a 2/3 rate coder.

These 3-bit codes are substituted for the original 2-bit words are represented by symbols with 8 combinations. Therefore, there are total eight symbols (-7, -5, -3, -1, 1, 3, 5, 7 scaled voltage levels) [82]. The symbol rate is 10.76 MHz [89].

Then two types of synchronization signals are embedded into the data, i.e. the segment sync and the field sync, as shown in Figure 4.4. The segment sync is a four-symbol sequence (+5, -5, -5, +5) and appears at the beginning of each data segment of 832 symbols. The field sync is a long sequence of +5 and -5 symbols and appears at the beginning of each field block containing 313 segments. Figure 4.5 shows the contents of the field sync. The two sync signals provide the carrier recovery and symbol timing recovery in the analog DTV receiver. However, in our work, only the field sync is used for synchronization. The reason that segment sync is not used lies in the severe multipath present at our antenna. The multipath distorts the shape of the segment sync, and the segment sync itself would be too short to accurately determine the symbol sampling time. Furthermore, segment sync is used to identify each data segment for decoding, but decoding is not necessary for our application.

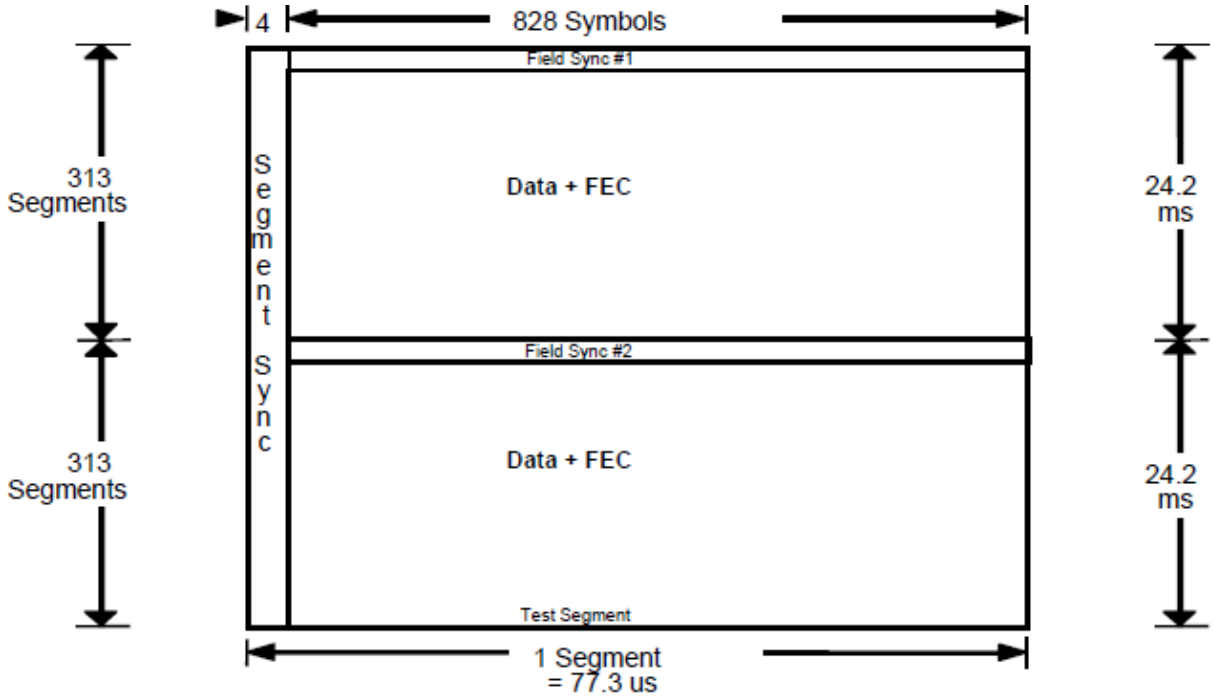


Figure 4.4. Data structure of DTV data frames [82].

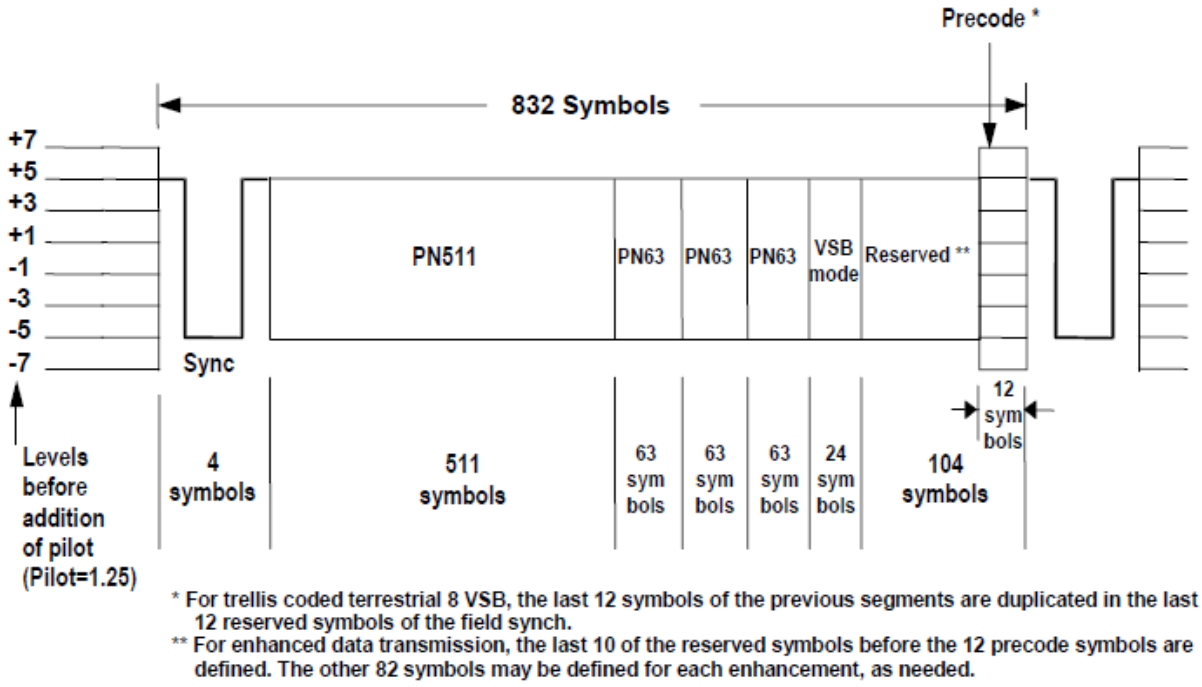


Figure 4.5. VSB Data Field Sync [82].

After the data encoding a DC offset of 1.25 is added to the signal, effectively creating a pilot tone [82, 90]. This offset signal is shifted to the intermediate frequency and filtered with a root raised cosine filter for vestigial sideband (VSB) modulation. Finally, the signal is shifted to carrier frequency to be transmitted.

On the receiver side, there is another identical root raised cosine filter [91]. In the concatenated transmitter and receiver, the resulting transition regions have a raised cosine shape, as shown in Figure 4.6. The raised cosine filter is a Nyquist filter with 5.38 MHz bandwidth, and it completely removes the lower RF sideband.

The frequency response of the raised cosine filter is

$$H(f)=\begin{cases} T, & |f|<\dfrac{1-\beta}{2T} \\ \dfrac{T}{2}\left\{1+\cos\left[\dfrac{\pi T}{\beta}\left(|f|-\dfrac{1-\beta}{2T}\right)\right]\right\}, & \dfrac{1-\beta}{2T}<|f|<\dfrac{1+\beta}{2T} \\ 0, & \text{otherwise} \end{cases} \tag{4.45}$$

where β the roll-off factor (0< β <1) that determines the bandwidth of the spectrum. T is the reciprocal of the symbol-rate. The impulse response of such a filter is given by:

$$h(t) = \text{sinc}\left(\frac{t}{T}\right)\frac{\cos\left(\frac{\pi\beta T}{T}\right)}{1-\frac{4\beta^2 t^2}{T^2}} \tag{4.46}$$

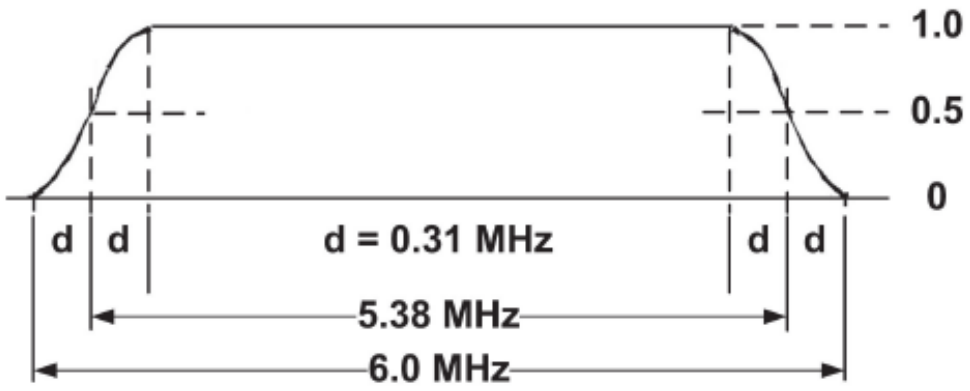


Figure 4.6. Overall VSB transmitter plus receiver linear amplitude response versus frequency (concatenation of linear phase root raised cosine Nyquist filters).

4.2.2 *Synchronization Challenges*

Conventional DTV demodulators may use analog phase-locking techniques to align the DTV waveform for symbol extraction [82, 89, 92]. However, our digitizer and FPGA require synchronous clocking to function, and the FPGA code is sensitive to clock rate, so that it cannot be easily changed. Furthermore, simultaneous detection of DTV and (say) GPS would create conflicting demands for clock phase and frequency control. So, we have developed a completely digital process to demodulate the 8VSB signal. We also deal with clock frequency drift using numerical methods, and thus relax requirements on clock stability.

The signal used here for demodulation comes from a Seattle area DTV station on channel 31 (KONG, pilot tone at 572.25 MHz). It is collected by the receiver then downconverted to baseband and finally downsampled to about 12 MSPS, with an output bit depth of 8, using a mode from Chapter 3.

The universal digital demodulation process involves applying a root Nyquist filter as seen in Figure 4.6 [82] [93], then filtering out the ground clutter and noise, and finally rounding each sample to one of the 8 quanta. However, the synchronization is a challenge because the clock is

not identical and not locked to the transmitter clock and there is no handy physical circuit for synchronization. Therefore, the digital solutions for these challenges involve changing sampling rate to DTV standard symbol rate, synchronizing sampling time, aligning the field sync with a simulated reference field sync. We will discuss the process in details in sequence.

4.2.3 *Digital Demodulation Process*

There are 9 steps in the digital demodulation process:

(1) Estimate the receiver clock because sampling rate is not accurate due to the fixed point operation and slight clock offset and drift.

(2) Numerically change the sampling rate to the DTV standard symbol rate by using spline interpolation.

(3) Synchronize carrier frequency and remove the pilot tone

(4) Apply root-raised cosine filter identical to the filter on the transmitter side.

(5) Identify field sync in the data

(6) Synchronize the carrier phase using phase compensation

(7) Equalize the signal (with the help of DTV field sync, to produce a filter)

(8) Convert the baseband IQ SSB (Single Side Band) signal to real-valued DSB (Double Side Band) signal using the property of the analytic signal.

(9) Round the filter outputs to the eight-level symbols.

Each of the processed is addressed in more details below.

(1) Estimate the receiver clock because it drifts slowly from the specified frequency.

In order to recover the real symbol value, the data rate should be changed to the DTV standard symbol rate (10.76 MHz) from the raw data rate (about 14 MHz). However, the raw data rate is not constant because the frequency of the receiver master clock slowly drifts with respect to the DTV clock. The data rate is estimated by making use of the autocorrelation of the data and the field sync which repeats every 22.4 milliseconds. From the autocorrelation, the field sync appears as repeated peaks, as shown in Figure 4.7. One second's worth of autocorrelation lags is computed

to estimate the frequency offset to an accuracy of 1 Hz. The number of the samples for the first 40 peaks is counted, and is compared to the number (832×313×40 samples) at DTV standard symbol rate of 10.76 MHz. The simultaneous data rate can be converted from the number of counts. If necessary, we can further estimate the receiver master clock because the data rate is a fraction of the clock rate.

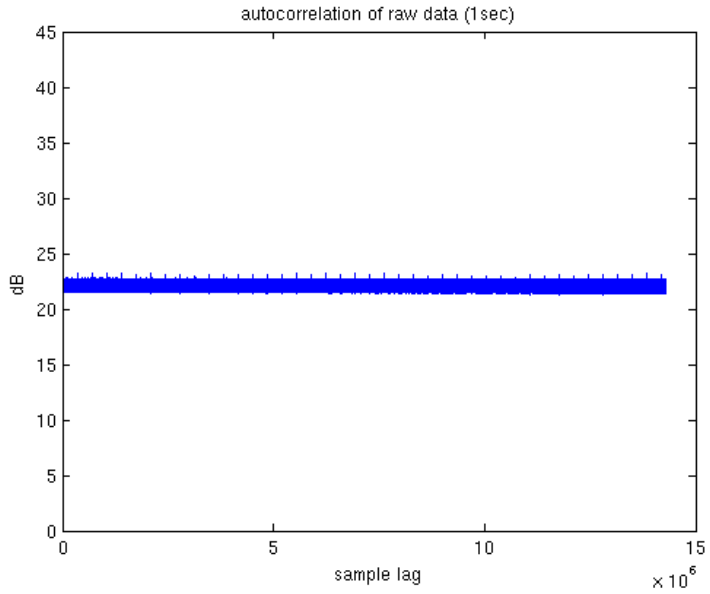


Figure 4.7. Magnitude of the autocorrelation of 1-second worth of received data.

(2) Numerically change the sampling rate to the DTV standard symbol rate by using spline interpolation.

Because the sampling change ratio is not an integer, a regular upsampling or downsampling that uses a combination of filtering and sampling rate change would not work. Polynomial interpolation is thus introduced to cope with the fractional rate change. Spline interpolation [94] is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. It doesn't require the two sampling rates to be integers or to have a small Least Common Multiple for upsampling or downsampling. It has more computation requirement than other polynomial interpolation methods, but it also has relatively smaller interpolation errors. When implemented in real time where computation speed is concerned in the future, other methods can be explored and compared and replace spline interpolation.

(3) Synchronize the carrier frequency and remove the pilot tone.

The pilot tone was originally a DC offset before the signal is modulated to the carrier frequency. Therefore, we can remove the pilot tone by shifting it to zero frequency and subtracting the sample mean from the data. Figure 4.8 (a) and Figure 4.8 (b) show the spectrum before and after the pilot tone is shifted to zero frequency. Figure 4.8 (c) shows the spectrum after the pilot tone is removed and shifted back to -2.69 MHz. The method works well when the pilot tone is not under severe selective fading. Fortunately, we have four antennas and we only need one for demodulation. We also correct the carrier frequency mismatch by the shift. The inaccuracy of the central frequency of the digital mixer is due to the finite precision of the digitally controlled oscillator and the unstable clock which the central frequency is derived from at this time. If left uncorrected, the signal shows unwanted phase change as a function of time.

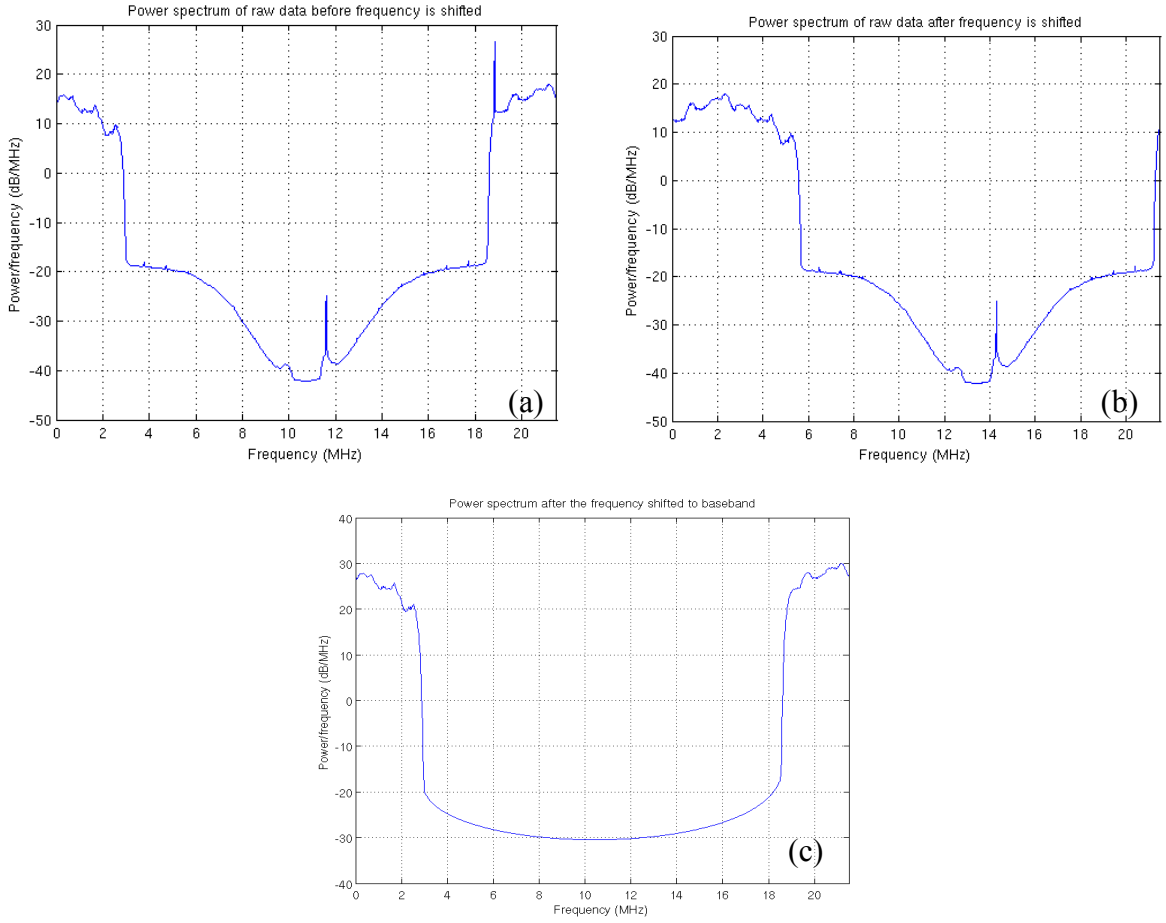


Figure 4.8. Power spectrum of the raw data before (a) and after (b) the pilot tone is shifted to zero frequency; (c) Power spectrum of the data after the pilot tone is removed.

(4) Apply root-raised cosine filter identical to the filter on the transmitter side.

After shifting back to baseband, the signal is applied with a root raised cosine filter which is identical to the filter on the transmitter side. The two filters form a raised-cosine Nyquist filter. The roll-off factor (beta) is 0.115 and the cutoff frequency is 2.69 MHz [82]. Unlike the raised-cosine filter, the impulse response is not zero at the intervals of $\pm T_s$. However, the combined transmit and receive filters form a raised-cosine filter which does have zero at the intervals of $\pm T_s$.

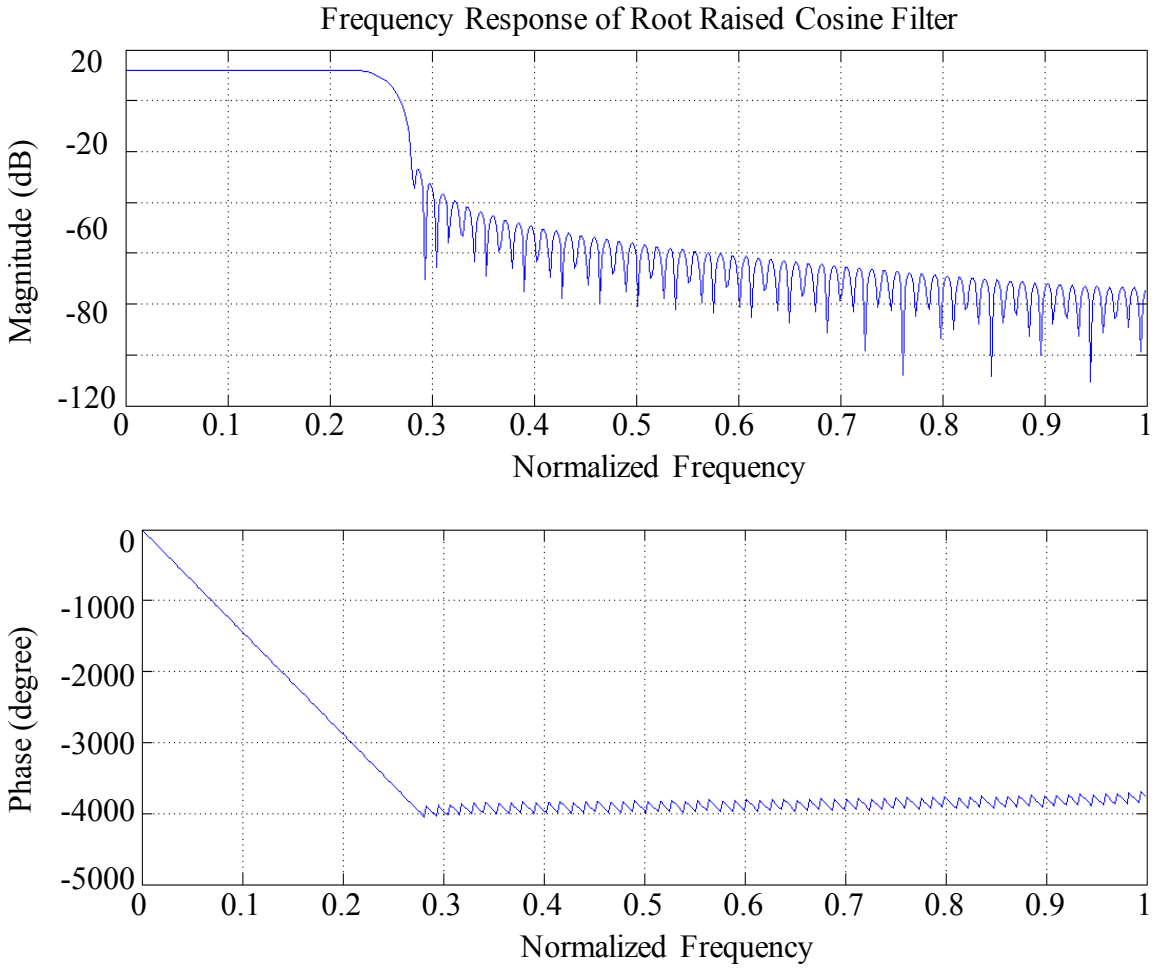


Figure 4.9. The frequency response of the root raised cosine filter.

(5) Identify field sync in the data

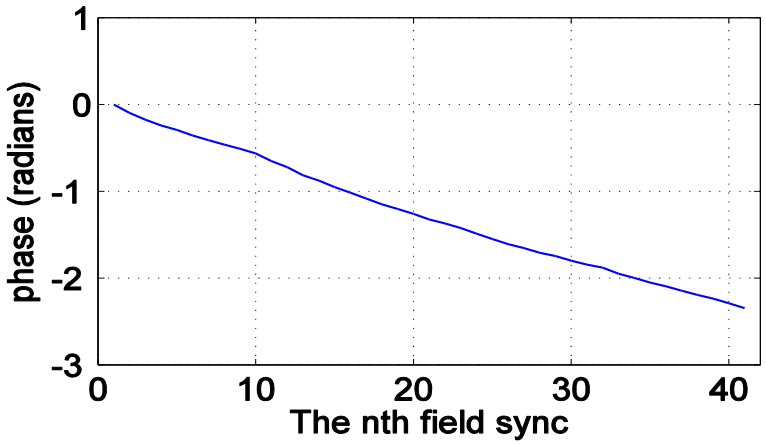
Because Nyquist filter would only provide desired SNR at the symbol sampling time of the transmitter, we need to locate the symbol sampling time for each sample as precisely as possible. Slightly misalignment will degrade SNR. Therefore, the identified field syncs in the data would be used together with a simulated field sync to recovery the symbol sampling time. Furthermore, the aligned field sync will be used as the input in the adaptive filter in the next two steps.

First, a simulated field sync which is composed of 832 known symbols is filtered with a raised cosine filter and is shifted to the baseband to be as a reference. The reference and the data are both upsampled by 10 for better timing resolution. Then the cross correlation is computed. The first sample of each field sync in the data is located from the peak of the correlation. Finally, each field sync block is downsampled back to 10.76 MHz.

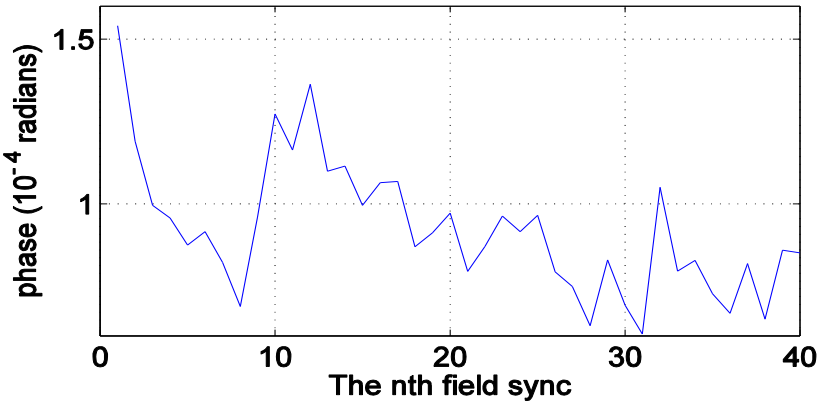
(6) Synchronize the carrier phase using phase compensation

Although the carrier frequency and sampling rate is synchronized at step 2) and 3), a small constant phase drift is still observed across the correlations between adjacent field syncs in the data, as shown in Figure 4.10 (a). A few components could cause it: slow drift of the transmitter sampling time (clock oscillator) or carrier frequency (transmitter analog mixer); slow drift of the receiver sampling time (receiver clock oscillator) and the receiver digital down mixer. Because the center frequency of the receiver mixer is a fraction of the receiver clock, it changes with the clock oscillator. Any variation of these components could result in the phase drift. As the phase drift has critical effect of the success of the adaptive filter algorithm to remove the multipath and direct path in the next step, we added phase compensation before adaptive filtering. The phase is compensated by multiplying $\exp\left(-j\,\Delta\varphi/(832\times313)\cdot t\right)$ for each field block, where $\Delta\varphi$ is the phase difference between two adjacent field sync, and t is time in seconds.

Figure 4.10 show the phase of the correlation between the simulated reference and 40 field syncs in the data before and after phase correction.



(a) The phase before phase correction



(b) The phase after phase correction

Figure 4.10. The phase difference of the cross correlation between the 40 field syncs in the data and reference field sync before (a) and after (b) the phase compensation. Time interval between two adjacent field syncs are 24.2 millisecond. Therefore, total 40 field syncs of the x axis correspond to a time duration of one second. (a) indicates the phase drifts about 2.3 radians in one second. Notice the large change in the scales for phase in (a) and (b).

(7) Equalization (with the help of DTV field sync, to produce a filter)

Several different methods, such as LMS (Least Mean Square), Wiener Hopf method, RLS (Recursive Least Square), are examined [85]. The results show that all methods have good convergence in the low noise environment, and with similar small residual signal. The input training signal is the field sync in the data, and the desired output is the simulated field sync

reference. The residual signal from which the direct and multipath have been removed has a typical power of 2% of the original signal power.

Figure 4.11 shows the field sync distribution on the complex plane before and after filtering. The value of the samples spread before filtering represents the signal with additive multipath and noise; with the adaptive filter algorithm it converges to the two quanta of [-5, +5] as expected.

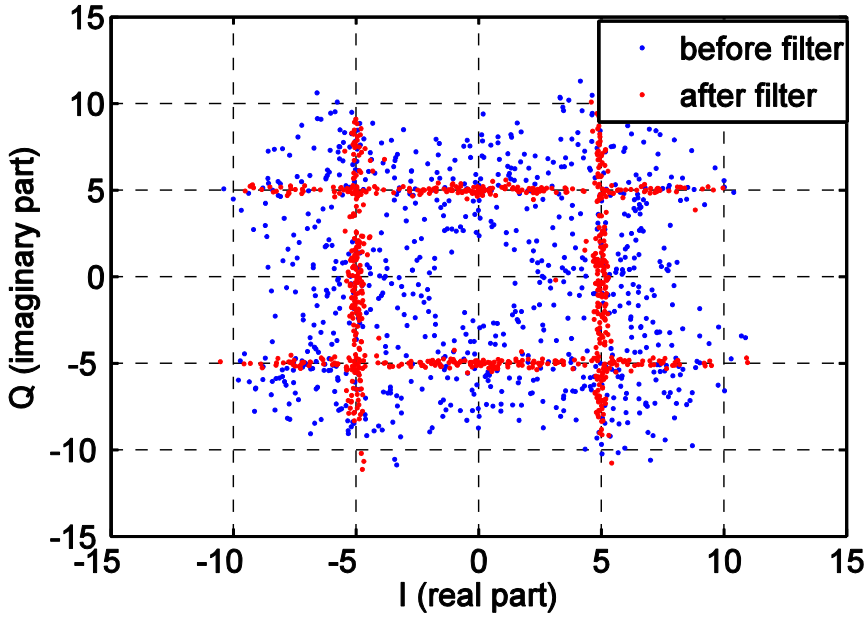


Figure 4.11. The field sync sample distribution on the complex plane before (blue) and after (red) the application of the adaptive filter algorithm.

The filter produced with the algorithm is applied to the entire field block of IQ data (each filed block has 832x313 symbols). Figure 4.12 shows the constellation of the IQ samples. Both the real and imaginary part of the samples are distributed in the eight quanta. Because the baseband signal is phase shifted by one of the following: +j, -j, +1, -1, either real or imaginary part must have the same the eight quanta.

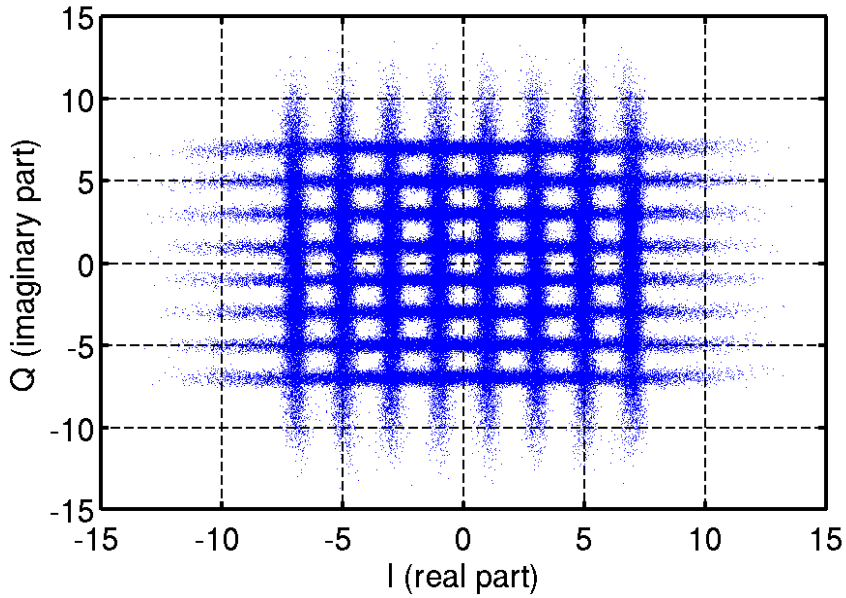


Figure 4.12. The constellation of the IQ samples of the DTV data (one field block) after the application of the adaptive filter algorithm.

(8) Convert the baseband IQ SSB (Single Side Band) signal to real-valued DSB (Double Side Band) signal using the property of the analytic signal.

The IQ data is converted to real value by frequency shifting +2.69 MHz and extracting the real part. Because the SSB signal is analytic signal, the imaginary part of the signal is the Hermite transform of the real-valued data, and can be discarded [95].

(9) The outputs of the filter are rounded to the eight-level symbols.

4.2.4 *Error Rate Estimate*

When the fluctuation on each symbol is less than one quantum, the sample can be correctly classified to one of the 8-level symbols. The error rate for each field block is estimated from the histogram of the values.

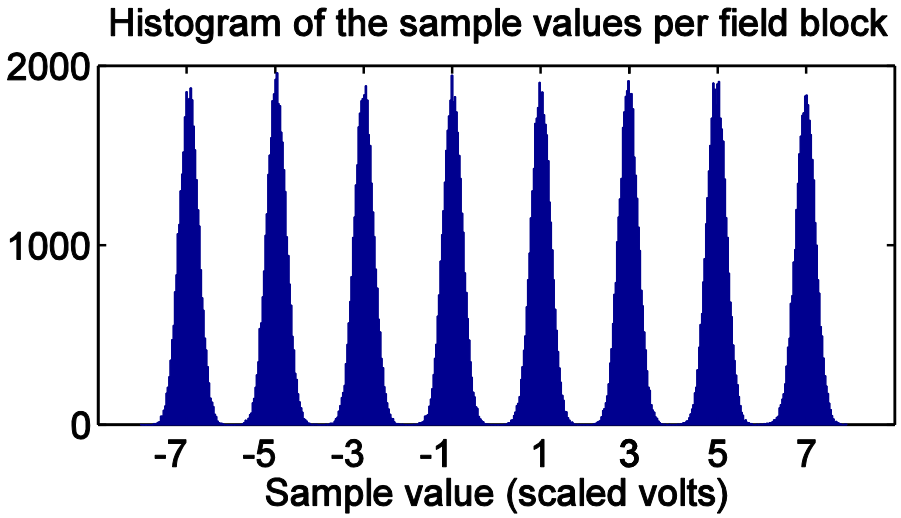


Figure 4.13. Histogram of the sample values, showing their tight clustering near the 8 symbols (-7, -5, -3, -1, 1, 3, 5, 7).

Figure 4.13 shows the histogram of the filter output of the entire field block. From Figure 4.13, we observed an approximately Gaussian distribution around each symbol, and the deviation is much less than 1 quantum. The number of samples with large deviation from the symbols is very small. We modeled a Gaussian distribution for each of the 8-level symbols with the measured means and deviations. The thresholds are halfway between the symbols. Then the error probability of each symbols can be calculated by the following methods:

$$p_i = 1 - \mathrm{erf}\left(\frac{1}{\sqrt[2]{2}\sigma_i}\right) \qquad i = \pm 1, \pm 3, \pm 5, \pm 7 \tag{4.47}$$

Where in Equation 4.47 i is the one of the 8-level symbols, 1 on the numerator is from the threshold beyond which the symbol is mis-classified, and σ_i is the standard deviation.

Each symbol is uniformly distributed, so the total error probability is

$$p_{total} = \frac{1}{8}\sum_i p_i \tag{4.48}$$

Table 4.1. Estimated Errors of One Field Block

Symbols	**±1**	**±3**	**±5**	**±7**
Number of estimated symbol errors per field block	0.4	0.4	0.5	0.7
Number of total estimated symbol errors	2 samples			

Table 4.1 lists the number of estimated symbol error of one field block (832x313 samples). The estimated total number of symbol errors, or the error rate, is only 2 per over 2×10^5 samples, indicating good performance of the digital demodulation algorithm. The number of errors for low-magnitude symbols may be smaller than the number for the ±7 symbol, mainly because the high-magnitude symbol is truncated before transmission to avoid excessive signal power required of the transmitter [82, 89].

4.3 Direct Path and Multipath Removal

The recovered signal is rounded to the nearest symbol and is then converted to baseband IQ form by VSB modulation (with a raised cosine filter and frequency shift). The baseband signal serves as the reference input signal for the radar, and the original received signals arise from the multipath channels to each antenna. With the adaptive filter algorithm, such as LMS, the direct path and non-moving and low-Doppler frequency ground clutter are removed from the received signal. The residual signal ideally only contains the noise and targets with Doppler frequencies.

With LMS algorithm, the impulse response of the channel can be estimated at the same time. Figure 4.14 (a) shows the impulse response of the channel with 1000 filter taps. Each tap represents the back scattering range resolution of c/fs/2 = 300Mm/us/10.76MHz/2 = 15 meters. The filter extends over a total of 15 km, beginning 3 km before the strongest signal, because the prompt direct path signal may not be the strongest signal to arrive. Most power in the multipath comes from the nearest ranges. As the distance increases, the ground clutter power diminishes. Figure 4.14 (b) shows the first few taps which represent the strong ground clutter near the direct path. Figure 4.14 (c) shows an expanded view of the coefficients in order to reveal the details of the low power ground clutter.

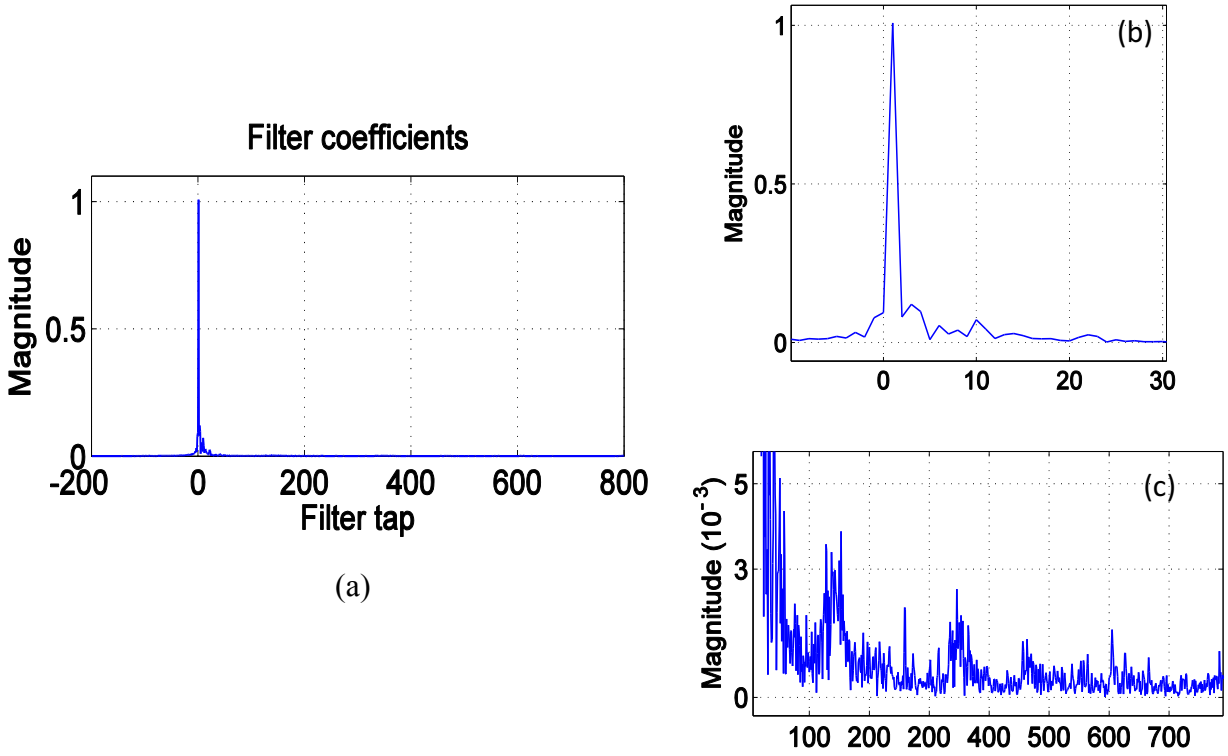


Figure 4.14. Magnitude of filter coefficients of the direct path and multipath. (a) all 1000 taps; (b) expanded view of the largest the taps; (c) expanded view of the ground clutter ranges.

4.4 Dynamic Range Increase in Ambiguity Function

4.4.1 *Cross Ambiguity*

The purpose of demodulation is not to recover the TV video signal, but to improve the detectability of the target in the radar system. With the recovered transmission DTV signal, we can improve the dynamic range of ambiguity function [81, 96, 97] by removing the direct and multipath. The range resolution of DTV signals is better than FM signals due to the large bandwidth of the DTV signal, and DTV signals usually have higher average power than the FM signals.

To calculate the ambiguity function, we used 0.8 seconds of DTV data at a date rate of 10.76 MHz. The coherent integration length is 1024 and FFT length is 8192. The Doppler velocity ranges from -1.35 km/s to 1.35 km/s with resolution of 0.3 m/s per Doppler bin. The backscattering range resolution is 15 m per bin.

First, the self-ambiguity of the data is calculated. The theoretical signal to numerical noise floor is

$$SNR = 10log10(B\tau) = 10log10(6\ MHz \cdot 0.8sec) = 67\ dB \quad (4.49)$$

where B is the effective bandwidth of DTV. The self-ambiguity calculated with the data has a 0 dB normalized direct path power and an average noise floor of -67 dB, so the SNR is 67 dB, which is identical to the theoretical SNR.

Then, the DTV demodulation method is implemented, and thus we have both the recovered transmitted waveform (reference signal), and the residue which contains noise and target echoes. The cross ambiguity between the reference signal and residue is calculated. The result plot is shown in Figure 4.15. From the figure, we can see that the numerical noise floor is reduced to -100 dBc due to the removal of direct path and multipath signal.

4.4.2 *Target Detection*

In order to compare the detectability of with targets with different reflected powers, two artificial targets with some Doppler velocity are numerically added with one aA t -40 dBc and the other at -80dBc. The first target (-40 dBc) is observable in the self-ambiguity because its power is higher than the noise floor (-67 dB), but the second (-80 dBc) is invisible because its power is lower than the noise floor.

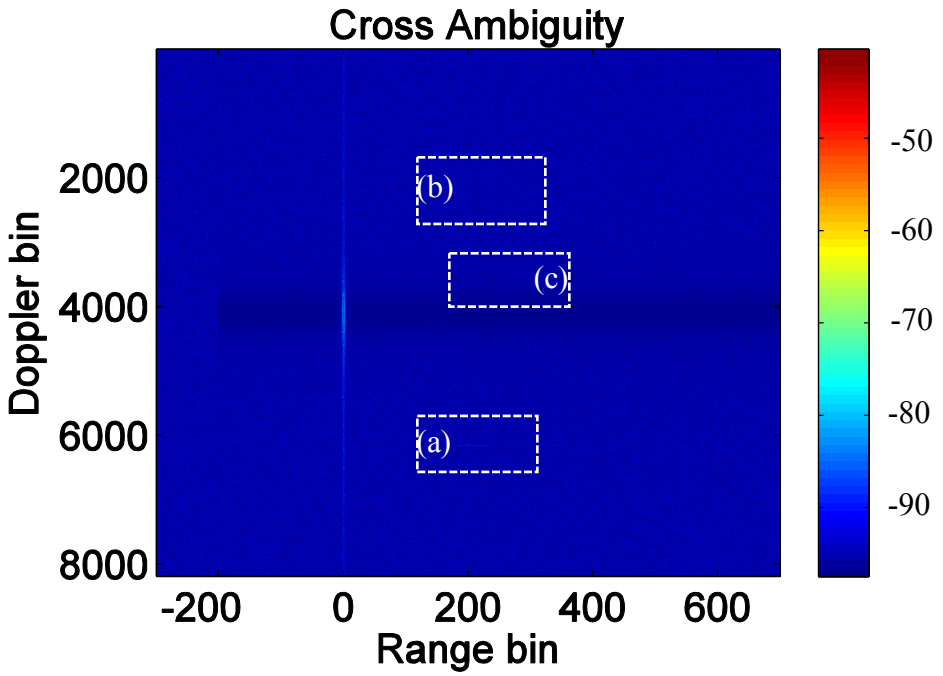


Figure 4.15. The ambiguity function of the residue and the reference signal is computed with number of samples of 0.8 second. areas of (a), (b) and (c) are zoomed in Figure 4.16 to show the targets.

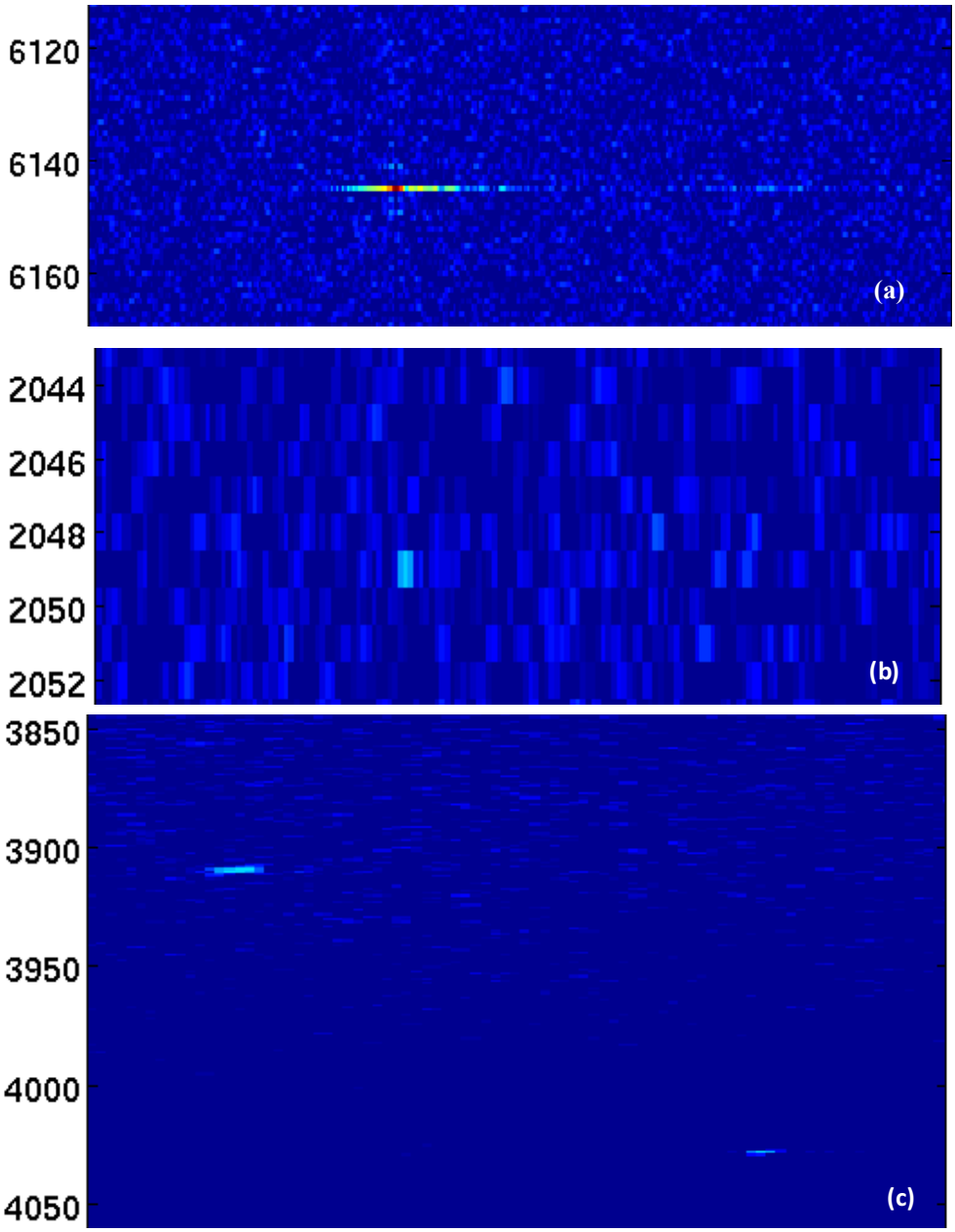


Figure 4.16. Expanded view of the ambiguity function in Figure 4.15. (a) expanded view of the -60 dB target; (b) expanded view of the -80 dB target; (c) expanded view of the two real targets.

After the noise floor is reduced to -100 dBc, the second artificial target can be observed at -80 dBc, as shown in Figure 4.16 (b), 13 dB above the residual clutter floor. Figure 4.16 (a) shows the first target at -40 dBc, 60 dB above the residual clutter floor.

Notice that the self-correlation at zero lag and the ground clutter are mostly removed. Also, some Doppler spread around zero lag remains because some signal from the direct path and strongest ground clutter is not completed removed by our implementation of the LMS algorithm. The numerical noise floor around zero Doppler range is relative lower than high Doppler range because the LMS algorithm is iterative and can adapt to remove lowly varying signals.

When we set the detection threshold at -82 dBc, we also detected two real targets from the cross ambiguity, in addition to the two artificial targets. Their powers are about -80 dBc as shown in Figure 4.16 (c), one at 60m/s at 2.6 km, and the other at 20m/s at 3.4 km. During 0.8 seconds, the targets' range migrations are 48 m and 16 m respectively, which are indicated in the ambiguity plot with range migration of 4 bins and 2 bins.

4.4.3 *Statistics of Clutter Floor*

In this section, we characterized the numerical clutter floor in the cross ambiguity. Figure 4.17 shows the histogram of the ambiguity noise floor (log magnitude). The noise floor is Rayleigh-distributed with sample mean at -98 dB.

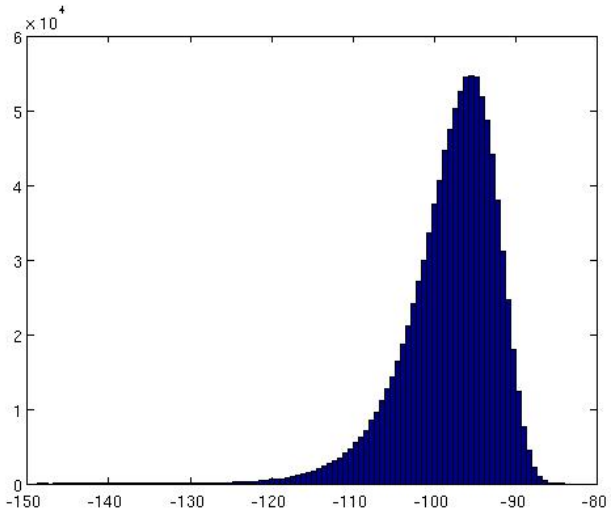


Figure 4.17. Histogram of noise power in the cross ambiguity.

For a given detection threshold T, we can compute the probability of false alarm and the probability of detection based on the noise power distribution. Suppose the noise or clutter power

distribution in Figure 4.17 is $f(x)$. In the present of a signal with power y, the signal pulse power will be distributed as $f(x-y)$. Therefore, the Probability of False Alarm (P_{fa}) is computed as follows:

$$P_{fa} = \int_{T}^{\infty} f(x)dx \tag{4.50}$$

Likewise the probability of Detection (P_d) is

$$P_d = \int_{T}^{\infty} f(x-y)dx \tag{4.51}$$

Based on Equation 4.50 and Equation 4.51, we can product False Alarm v.s. Detection plot for a given signal power or SCR (y in Equation 4.51). Probability of False Alarm v.s. Probability of Detection for for a set of SCRs (in dB). The plot indicates that when the probability of false alarm is low, the probability of detection is low, and vice versa. The curve moves towards to the upper left corner as SCR becomes larger, which indicates that large signal power tends to have a higher detection probability and lower false alarm probability.

Figure 4.18 shows the receiver operating characteristic for a single range Doppler cell. More elaborate detection algorithms are possible, such as declaring a detection if at least two of three adjacent cells exceed a threshold. Evaluation of alternative detection schemes is beyond the scope of this work. However, the interested reader may wish to examine [98, 99].

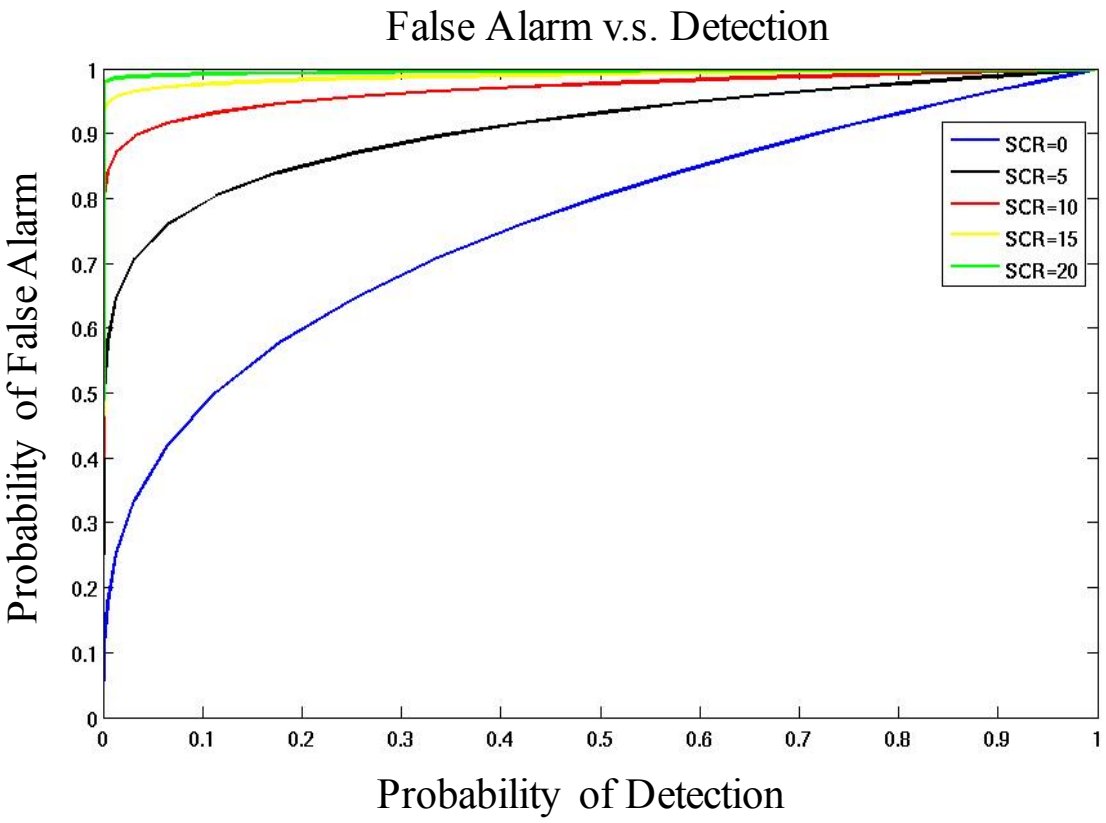


Figure 4.18. The receiver operating characteristic is the Probability of False Alarm v.s. Probability of Detection for for a set of SCRs (in dB). The plot indicates that when the probability of false alarm is low, the probability of detection is low, and vice versa. The curve moves towards to the upper left corner as SCR becomes larger, which indicates that large signal power tends to have a higher detection probability and lower false alarm probability.

4.5 LMS Influence of Removing Non-zero Doppler Target

As an adaptive filter, LMS is able to adapt as the communication channel changes, or remove the direct path and multipath with slow Doppler shift. Therefore, a moving target with slow Doppler shift might be removed or partially removed during LMS operation. A simulation is created to analyze the LMS influence on the target with slow Doppler shift. The target with the magnitude of 10% of the direct path with bistatic range of 1.4 *km* at different Doppler frequencies was added. The input baseband signal has a sampling rate at 21.52 MHz (twice DTV standard sample rate)

and pilot tone at DC removed. The data is normalized as the input of LMS algorithm. A corresponding frame sync, whose symbols are well known, is created as a reference signal for LMS training purpose. The LMS algorithm applied has 1101 taps with update factor of 0.1. As the some ground clutters have low Doppler frequency such as traffic on the highway, leaves moving of trees nearby. propagation path of ground clutter reflection is not stationary due to n the long term is The value of update factor is chosen such that the LMS filter adapts the multipth

In the ambiguity calculation, we coherently integrate 2,000 samples at 21.52 MHz, and therefore the Doppler frequency is 10.76 kHz or range (-5.38 kHz, 5.38 kHz), and the total time and corresponding SNR for one frame of ambiguity plot is

$$\tau = \frac{512}{10.76kHz} = 0.0476sec \tag{4.52}$$

$$SNR = 10log10(B\tau) = 10log10(5.38MHz \cdot 0.0476sec) = 54.9dB \tag{4.53}$$

For a 512-point FFT ambiguity function, the Doppler frequency resolution and Doppler velocity resolution is

$$\Delta f_d = \frac{10.76kHz}{512} = 21\ Hz \tag{4.54}$$

$$\Delta v_d = \frac{\lambda}{2}\Delta f_d = \frac{0.52m}{2}21Hz = 5.46\ m/s \tag{4.55}$$

The bistatic range per range bin is

$$\Delta r = \frac{c/2}{21.52\ MHz} = \frac{300\ Mm/s}{2 \cdot 21.52\ MHz} = 7\ m \tag{4.56}$$

The bistatic range resolution is

$$\Delta r = \frac{c/2}{6\ MHz} = \frac{300\ Mm/s}{2 \cdot 6\ MHz} = 25\ m \tag{4.57}$$

Figure 4.19 below shows the self-ambiguity plot before applying the LMS algorithm, in other words, the ambiguity with direct path and ground clutter in it. Horizontal axis is the bistatic range with about 7 meter/bin. Vertical axis is Doppler shift with zero Doppler shift at the 257th bin. The peak at the 200th bin is the direct path power, which shows at 9 dB, and the noise floor is at -44

dB. Therefore the signal to noise ratio (SNR) is 53 dB. It's a little less than the theoretical value because the additional noise contributed by the ground clutter and the aircraft.

LMS algorithm would remove direct path, ground clutters and low Doppler frequency targets, and the residue is the signal that only contains medium and high Doppler frequency targets. Then we calculated the cross ambiguity between the residue and transmitted waveform. Figure 4.20, Figure 4.21, Figure 4.22 and Figure 4.23 show the cross ambiguity after LMS algorithm, each with target at Doppler bin of 200, 243, 253 and 257 individually. Table 4.2 lists the noise floor and target power of each figure.

The noise floor dropped significantly to -67 dB after the direct path and multipath are removed with LMS algorithm, as shown in Table 4.2. Therefore, after the noise floor dropped, the target located at the 200th Doppler bin appears in the ambiguity plot (see Figure 4.20). As the target gets closer to zero-Doppler range, its power is decreased from -11 dB of true power to -35 dB at the 253th bin, and further to some value below the noise floor when the target is at 257th Doppler bin (the zero-Doppler shift). This is because of the LMS's ability to remove low Doppler frequency echoes.

It also shows that as the target power gets weaker towards zero-doppler range, the noise floor is correspondingly dropped. When the target power is strong, it can significantly rise the numerical noise floor.

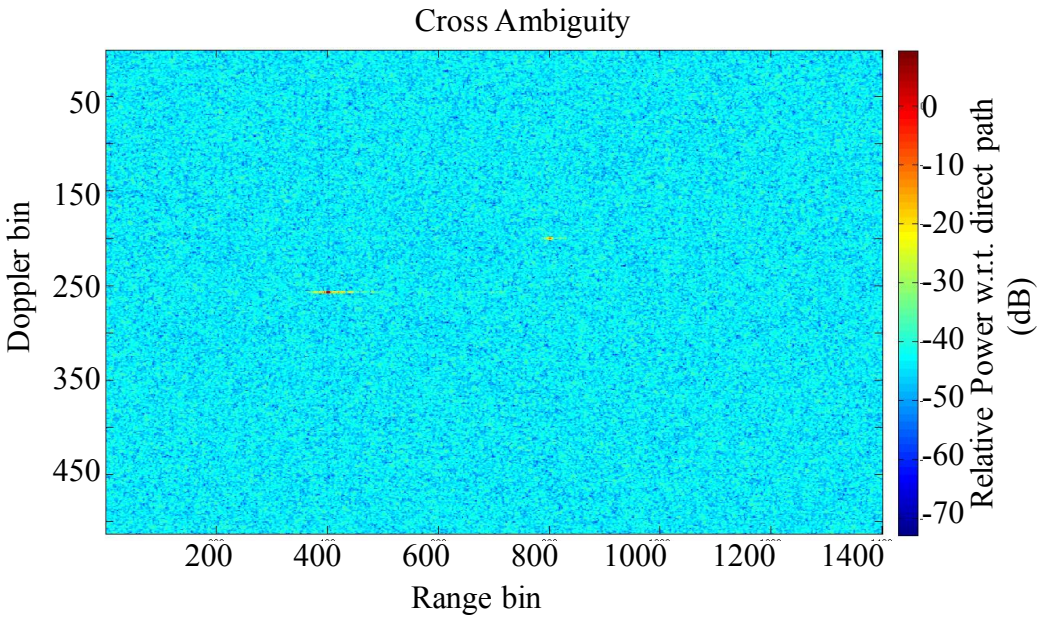


Figure 4.19. Cross ambiguity between received signal and LMS-extracted transmitted waveform before applying the LMS algorithm with the simulated target at Doppler bin of 200.

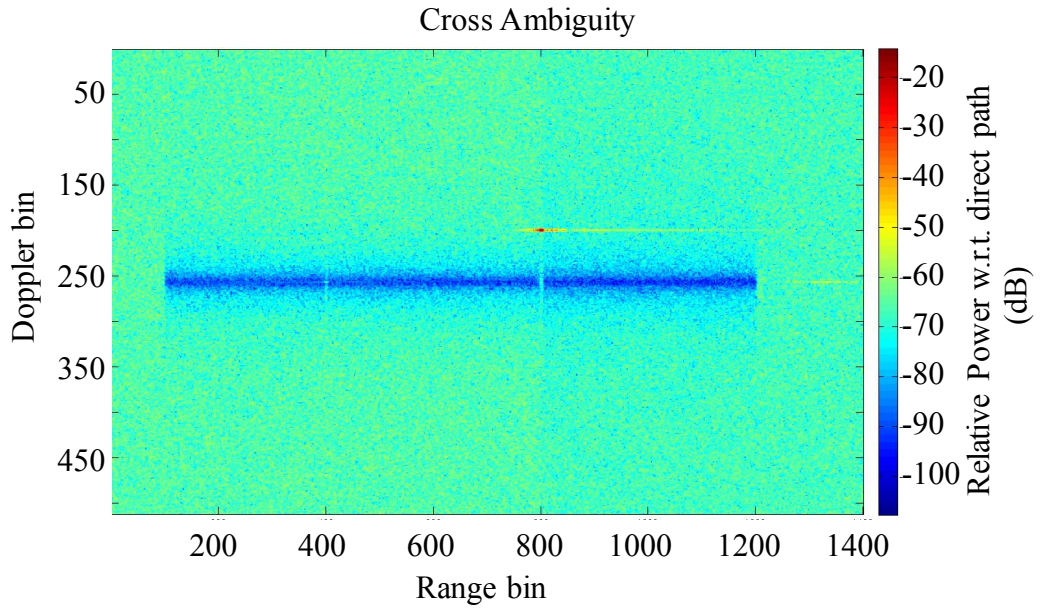


Figure 4.20. Cross ambiguity between residue signal and LMS-extracted transmission signal with the simulated target at Doppler bin of 200.

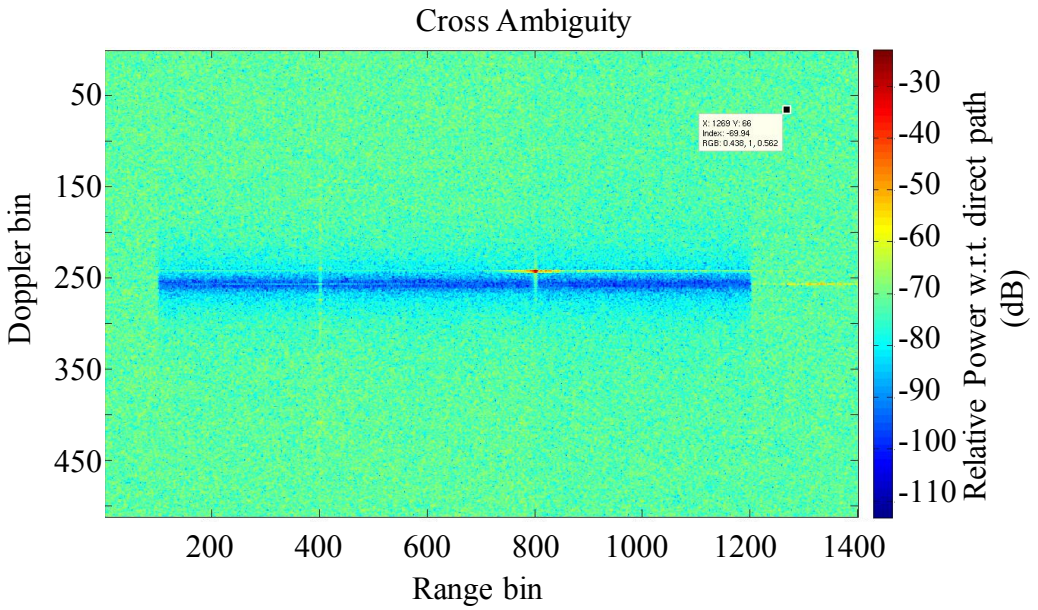


Figure 4.21. Cross ambiguity between residue signal and LMS-extracted transmitted waveform with the simulated target at Doppler bin of 243.

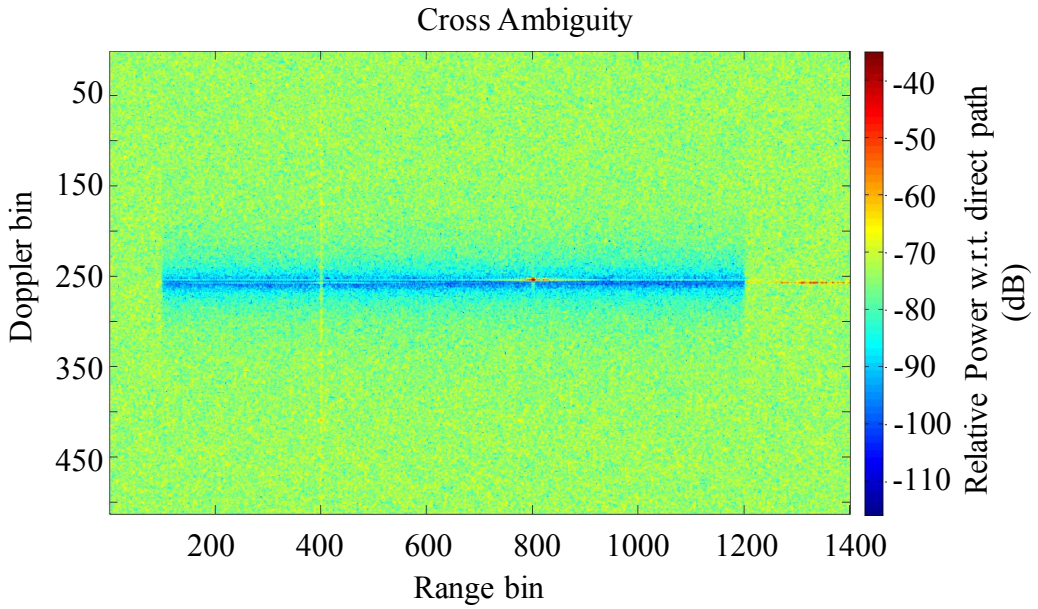


Figure 4.22. Cross ambiguity between residue signal and LMS-extracted transmitted waveform with the simulated target at Doppler bin of 253.

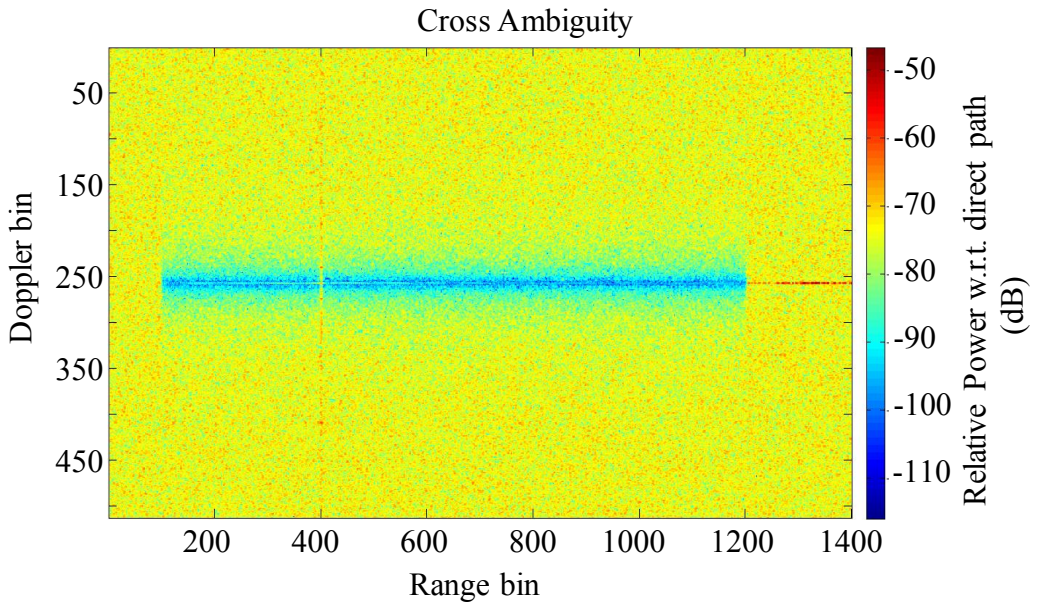


Figure 4.23. Cross ambiguity between residue signal and LMS-extracted transmitted waveform with the simulated target at Doppler bin of 257.

Table 4.2. Noise Power and Target Power in Figure 4.19 - Figure 4.23

Ambiguity function	**Target Location**	**Figure**	**Noise power**	**Target power**
Self-ambiguity function before applying LMS algorithm	200th Doppler bin	Figure 4.19	-44 dB	-11 dB (actual power)
Cross ambiguity function after applying LMS algorithm	200th Doppler bin	Figure 4.20	-67 dB	-14 dB
Cross ambiguity function after applying LMS algorithm	243th Doppler bin	Figure 4.21	-73 dB	-23 dB
Cross ambiguity function after applying LMS algorithm	253th Doppler bin	Figure 4.22	-75 dB	-35 dB
Cross ambiguity function after applying LMS algorithm	257th Doppler bin	Figure 4.23	-75 dB	N/A

4.6 SUMMARY

In this chapter, we have developed a completely digital process to demodulate the 8VSB signal. With a nearly ideal demodulated reference signal, we demonstrated 98 dB instantaneous dynamic

range with about 0.8-second coherent processing. If needed, 100 dB dynamic range can be reached by 1.3-second coherent processing.

Chapter 5. CONCLUSIONS AND FUTURE EXPERIMENTATION

5.1 CONCLUSIONS

In this document, we describe the development of a new high speed passive radar system and dynamic range improvement of passive radar with use of Digital TV as the illumination source. The new receiver is equipped with two 2.5/5GSPS samplers and Xilinx Virtex 6 FPGA, and is developed to have the following performance:

1. High speed samplers and digital signal processors offer greatly relaxed of requirements for analog filters, and thus the system has great linearity performance.
2. For each RF input, the new receiver is capable of simultaneously extracting multiband channels (up to 3 frequency channels) from the fast sampler and FPGA processing ability.
3. Down conversion is performed numerically in the FPGA. Therefore, the two 90-degree phase-shifted oscillators can be designed to achieve nearly perfect quadrature in amplitude and phase over the entire frequency range.
4. The receiver is straightforward to operate once the FPGA programming has been performed. Our current FPGA code design also allows the tuning frequencies to be changed in a few milliseconds without halting the data acquisition system.
5. When working at 5 GHz, it covers most commercial broadcasts, which makes more waveforms available for passive radar application.

We further explored the passive radar application with the new receiver. The DTV signals are studied in the passive radar application for its large average power, large bandwidth compared to FM, and noise like spectrum. As passive radar is signal-to-clutter ratio limited, we proposed the digital demodulation method of DTV signals to extract the transmitted signal and thus improve the dynamic range to over 100 dB. The weak echoes from small targets can be observed with the significantly reduced noise floor.

5.2 FUTURE DEVELOPMENT

In this section we briefly explore ideas for upcoming research with this new system.

5.2.1 *System Performance Improvement*

The current system hardware is quite simple as our receiver is able to tolerate a fair amount of imperfection. Furthermore, it is able to optimize each analog element, such as the impedance and frequency response of the antenna, antenna array platform and geometry, physical shielding of the antenna from the exposure to direct path, site selection of the receiver for better reception.

In the market, there are also higher speed ADCs as the advances of the digital electronics. We can explore these ADCs with less bit resolution.

5.2.2 *Target Location Exploration*

For detection application, we have demonstrated 98 dB dynamic range of 0.8 sec coherent integration. We have done some preliminary AoA estimation for target location with rough spatial resolution, and this work can be carried on in the further to achieve better solution by accurate array calibrations.

With our multi-antenna receiver, we can also explore other location algorithms, such as beamforming, interferometry, etc. and it is also possible to explore TDOA based on the existing DTV transmitter network.

5.2.3 *Waveform Exploration*

As our receiver has a large unaliased bandwidth up to 2.5 GHz, it is able to explore more waveforms in the further, such as FM broadcast, Wi-Fi and GPS, etc. It is able to improve the frequency resolution and range resolution by using multiple waveforms.

5.2.4 *Aurora Observations*

We have in the past observe aurora with FM broadcast. It would be interesting to know what the ambiguity of aurora looks like with DTV signals.

BIBLIOGRAPHY

[1] F. Colone, D. W. O. Hagan, P. Lombardo, and C. J. Baker, "A Multistage Processing Algorithm for Disturbance Removal and Target Detection in Passive Bistatic Radar," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 45, no. 2, pp. 698-722, 2009.

[2] F. Colone, P. Falcone, C. Bongioanni, and P. Lombardo, "WiFi-Based Passive Bistatic Radar: Data Processing Schemes and Experimental Results," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 48, no. 2, pp. 1061-1079, 2012.

[3] J. Baniak, G. Baker, A. M. Cunningham, and L. Martin, "Silent Sentry passive surveillance," *Aviation week and space technology,* vol. 7, pp. 134-139, 1999.

[4] M. Cherniakov, D. Nezlin, and K. Kubik, "Air target detection via bistatic radar based on LEOS communication signals," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 149, no. 1, pp. 33-38, 2002.

[5] J. D. Sahr and F. D. Lind, "The Manastash Ridge radar: A passive bistatic radar for upper atmospheric radio science," *Radio Science,* vol. 32, no. 6, pp. 2345-2358, 1997.

[6] H. D. Griffiths and N. R. W. Long, "Television-based bistatic radar," *IEE Proc. F, Commun. Radar Signal Process.,* vol. 133, no. 7, pp. 649-657, 1986.

[7] H. D. Griffiths, A. J. Garnett, C. J. Baker, and S. Keaveney, "Bistatic radar using satellite-borne illuminators of opportunity," *92 International Conference on Radar,* pp. 276-279, 1992.

[8] H. D. Griffiths and C. J. Baker, "Passive coherent location radar systems. Part 1: Performance prediction," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 153-159, 2005.

[9] C. R. Berger, B. Demissie, J. Heckenbach, P. Willett, and S. Zhou, "Signal processing for passive radar using OFDM waveforms," *IEEE Journal of Selected Topics in Signal Processing,* vol. 4, no. 1, pp. 226-238, 2010.

[10] D. K. P. Tan, H. Sun, Y. Lu, M. Lesturgie, and H. L. Chan, "Passive radar using global system for mobile communication signal: theory, implementation and measurements," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 116-123, 2005.

[11] P. E. Howland, D. Maksimiuk, and G. Reitsma, "FM radio based bistatic radar," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 107-115, 2005.

[12] C. L. Zoeller, M. C. Budge, and M. J. Moody, "Passive coherent location radar demonstration," in *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No.02EX540)*, 2002, pp. 358-362.

[13] A. N. Morabito, M. G. Meyer, and J. D. Sahr, "Improved computational performance for distributed passive radar processing through channelised data," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 179-184, 2005.

[14] S. Stein, "Algorithms for ambiguity function processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 29, no. 3, pp. 588-599, 1981.

[15] M. G. Meyer, "Passive VHF Radar Interferometer Implementation, Observations, and Analysis," Master degree, EE Dept. , University of Washington, 2003.

[16] M. G. Meyer and J. D. Sahr, "Passive coherent scatter radar interferometer implementation, observations, and analysis," *Radio science,* vol. 39, no. 3, 2004.

[17] E. Kudeki and G. R. Stitt, "Frequency domain interferometry: A high resolution radar technique for studies of atmospheric turbulence," *Geophysical Research Letters,* vol. 14, no. 3, pp. 198-201, 1987.

[18] J. D. Sahr, "Ionospheric measurement," in *Advances in Bistatic Radar*: SciTech Publishing, 2007, pp. 193-211.

[19] S. J. Tingay *et al.*, "On the detection and tracking of space debris using the Murchison Widefield Array. I. Simulations and test observations demonstrate feasibility," *The Astronomical Journal,* vol. 146, no. 4, p. 103, 2013.

[20] M. I. Skolnik, *Radar handbook*, 3 ed. McGraw-Hill Education; 3 edition 1970.

[21] J. B. Hagen, *Radio-Frequency Electronics: Circuits and Applications*. Cambridge University Press, 1996.

[22] L. Vertatschitsch and J. D. Sahr, "High bandwidth, multi-purpose passive radar receiver design for aerospace and geoscience targets," PhD. Dissertation, EE Dept., University of Washington Seattle, 2013.

[23] M. G. Meyer, "Remote sensing of localized ion acoustic waves with multistatic passive radar. ," PhD Dissertation, EE Dept., University of Washington, 2006.

[24] F. D. Lind, "Passive radar observations of the aurora," PhD Dissertation, EE Dept., University of Washington, 1999.

[25] F. Berizzi, M. Martorella, D. Petri, M. Conti, and A. Capria, "USRP technology for multiband passive radar," pp. 225-229, 2010.

[26] C. Coleman and H. Yardley, "Passive bistatic radar based on target illuminations by digital audio broadcasting," *IET Radar, Sonar & Navigation,* vol. 2, no. 5, pp. 366-375, 2008.

[27] P. F. Howland, "Target tracking using television-based bistatic radar," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 146, no. 3, pp. 166-174, 1999.

[28] J. O. H. N. G. N. Henderson *et al.*, "ATSC DTV receiver implementation," *Proceedings of the IEEE,* vol. 94, no. 1, pp. 119-147, 2006.

[29] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE transactions on communications,* vol. 28, no. 11, pp. 1867-1875, 1980.

[30] . *USRP website* Available: https://www.ettus.com/

[31] M. M. FINN, Johnson, R. S., "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates," *RCA Rev.,* vol. 30, pp. 414-465, 1968.

[32] . *ATSC*. Available: http://atsc.org/

[33] J. E. Palmer, H. A. Harms, S. J. Searle, and L. Davis, "DVB-T passive radar signal processing," *IEEE transactions on Signal Processing,* vol. 61, no. 8, pp. 2116-2126, 2013.

[34] Z. Gao, R. Tao, Y. Ma, and T. Shao, "DVB-T signal cross-ambiguity functions improvement for passive radar," *2006 CIE International Conference on Radar,* pp. 1-4, 2006.

[35] R. Saini and M. Cherniakov, "DTV signal ambiguity function analysis for radar application," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 133-142, 2005.

[36] *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T)*, 2015.

[37] *A/53: ATSC digital television standard*. Available: http://atsc.org/standard/a53-atsc-digital-television-standard/

[38] W. Sun, J. D. Sahr, and T. Goodson, "Digital demodulation of DTV signals for passive radar application," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1-6.

[39] W. C. Barott and J. Engle, "Single-antenna ATSC passive radar observations with remodulation and keystone formatting," *2014 IEEE Radar Conference,* pp. 0159-0163, 2014.

[40] F. D. Lind *et al.*, "Intercepted signals for ionospheric science," *Radio Science,* vol. 48, no. 3, pp. 248-264, 2013.

[41] J. C. Foster and F. D. Lind, "Intercepted Signals for Ionospheric Science (ISIS) Instrumentation for a Small Radio Science Array," Intercepted Signals for Ionospheric Science2005.

[42] F. D. Lind, J. D. Sahr, and D. M. Gidner, "First passive radar observations of auroral E-region irregularities," *Geophysical research letters,* vol. 26, no. 14, pp. 2155-2158, 1999.

[43] J. D. Sahr and M. Meyer, "Opportunities for passive VHF radar studies of plasma irregularities in the equatorial E and F regions," *Journal of atmospheric and solar-terrestrial physics,* vol. 66, no. 17, pp. 1675-1681, 2004.

[44] M. G. Meyer, J. D. Sahr, and A. Morabito, "A statistical study of subauroral E-region coherent backscatter observed near 100 MHz with passive radar," *Journal of Geophysical Research: Space Physics,* vol. 109, no. A7, 2004.

[45] A. N. Morabito, J. D. Sahr, Z. M. P. Berkowitz, and L. E. Vertatschitsch, "Phase noise analysis in distributed coherent noise radar systems," in *2008 International Radar Symposium*, 2008, pp. 1-4.

[46] A. N. Morabito, J. D. Sahr, Z. M. P. Berkowitz, and L. E. Vertatschitsch, "Post-processing phase noise mitigation performance comparison in a coherent distributed passive radar system," in *2008 International Radar Symposium*, 2008, pp. 1-4.

[47] A. N. Morabito, "Estimation and mitigation of phase noise in a coherent distributed passive radar system," PhD. Dissertation, Electrical Engineering University of Washington, 2008.

[48] S. Weiwei, L. Vertatschitsch, and J. D. Sahr, "High speed computation for new digital VHF and UHF passive radar receiver," in *2013 US National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2013, pp. 1-1.

[49] L. E. Vertatschitsch and J. D. Sahr, "A new mobile passive radar designed for transmitters up to 1.5 GHz for detection of aerospace and geoscience targets," in *2013 US National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2013, pp. 1-1.

[50] . *ROACH*. Available: https://casper.berkeley.edu/wiki/ROACH-2_Revision_2

[51] . *KatADC*. Available: https://casper.berkeley.edu/wiki/KatADC

[52] W. M. Waters and B. R. Jarrett, "Bandpass Signal Sampling and Coherent Detection," *IEEE Transactions on Aerospace and Electronic Systems,* vol. AES-18, no. 6, pp. 731-736, 1982.

[53] J. Mitola, "The software radio architecture," *IEEE Communications Magazine,* vol. 33, no. 5, pp. 26-38, 1995.

[54] A. A. Abidi, "The Path to the Software-Defined Radio Receiver," *IEEE Journal of Solid-State Circuits,* vol. 42, no. 5, pp. 954-966, 2007.

[55] D. M. Akos and J. B. Y. Tsui, "Design and implementation of a direct digitization GPS receiver front end," *IEEE Transactions on Microwave Theory and Techniques,* vol. 44, no. 12, pp. 2334-2339, 1996.

[56] A. Brown and B. Wolt, "Digital L-band receiver architecture with direct RF sampling," in *Position Location and Navigation Symposium, 1994., IEEE*, 1994, pp. 209-216.

[57] D. Jakonis, K. Folkesson, J. Dbrowski, P. Eriksson, and C. Svensson, "A 2.4-GHz RF sampling receiver front-end in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 6, pp. 1265-1277, 2005.

[58] D. H. Shen, H. Chien-Meen, B. B. Lusignan, and B. A. Wooley, "A 900-MHz RF front-end with integrated discrete-time filtering," *IEEE Journal of Solid-State Circuits,* vol. 31, no. 12, pp. 1945-1954, 1996.

[59] S. Lindfors, A. Parssinen, and K. A. I. Halonen, "A 3-V 230-MHz CMOS decimation subsampler," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 50, no. 3, pp. 105-117, 2003.

[60] R. B. Staszewski *et al.*, "All-digital TX frequency synthesizer and discrete-time receiver for Bluetooth radio in 130-nm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 39, no. 12, pp. 2278-2291, 2004.

[61] K. Muhammad *et al.*, "A discrete time quad-band GSM/GPRS receiver in a 90nm digital CMOS process," in *Proceedings of the IEEE 2005 Custom Integrated Circuits Conference, 2005.*, 2005, pp. 809-812.

[62] Y.-C. Ho, R. B. Staszewski, K. Muhammad, C.-M. Hung, D. Leipold, and K. Maggio, "Charge-Domain Signal Processing of Direct RF Sampling Mixer with Discrete-Time Filters in Bluetooth and GSM Receivers," *EURASIP Journal on Wireless Communications and Networking,* vol. 2006, no. 1, p. 062905, 2006.

[63] . *GNU Radio*. Available: https://www.gnuradio.org/

[64] . *USRP X300*. Available: https://www.ettus.com/product/details/X300-KIT

[65] "Xilinx System Generator for DSP (Getting Started Guide) ", Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/sysgen_gs.pdf

[66] . *Speed Optimization with PlanAhead*. Available: https://casper.berkeley.edu/wiki/Speed_Optimization_with_PlanAhead

[67] . *Interacting with CASPER Instruments*. Available: https://casper.berkeley.edu/wiki/Interacting_with_CASPER_Instruments

[68] M. I. Skolnik, *Radar handbook*, 3rd ed. New York: McGraw-Hill, 2008, p. 1 volume (various pagings).

[69] . *ADC1x5000-8*. Available: https://casper.berkeley.edu/wiki/ADC1x5000-8

[70] . *Minicircuits Amplifier model ZKL-2R5+*. Available: http://www.minicircuits.com/pdfs/ZKL-2R5.pdf

[71] J. Scheer and J. L. Kurtz, *Coherent radar performance estimation*. Boston: Artech House, 1993, p. 446.

[72] W. A. Kester, *Data conversion handbook*, 1st ed. Newnes, 2005.

[73] H. Zumbahlen, *Linear circuit design handbook*, 1st ed. Newnes, 2011, p. 960.

[74] . *ADC datasheet* Available: http://www.e2v.com/products/ev8aq160-quad-8-bit-1-252-55-gsps-adc/

[75] "Enhancing ADC resolution by oversampling," Technical report Available: www.atmel

[76] . *KONG DTV station reception map from FCC data*. Available: https://www.fcc.gov/media/engineering/dtvmaps

[77] . *Keysight FieldFox*. Available: http://www.keysight.com/en/pcx-x205201/fieldfox-handheld-rf-and-microwave-analyzers?cc=US&lc=eng

[78] . *Xilinx DDS Compiler v4.0*. Available: https://www.xilinx.com/support/documentation/ip_documentation/dds_ds558.pdf

[79] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, 3rd ed. Upper Saddle River: Pearson, 2010, pp. xxviii, 1108 p.

[80] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE transactions on acoustics, speech, and signal processing,* vol. 29, no. 2, pp. 155-162, 1981.

[81] N. Levanon and E. Mozeson, *Radar signals*. Hoboken, NJ: J. Wiley, 2004, pp. xiv, 411 pages.

[82] A. Committee, "Atsc digital television standard-part 2: Rf transmission system characteristics," ed: December, 2011.

[83] R. Citta and G. Sgrignoli, "ATSC Transmission System: VSB Tutorial," 1997.

[84] M. H. Hayes, *Statistical digital signal processing and modeling*. New York: John Wiley & Sons, 1996, pp. xv, 608 p.

[85] S. S. Haykin, *Adaptive filter theory*, 4th ed. Upper Saddle River, N.J.: Prentice Hall, 2002, pp. xvi, 920 p.

[86] C. J. Baker, H. D. Griffiths, and I. Papoutsis, "Passive coherent location radar systems. Part 2: Waveform properties," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 152, no. 3, pp. 160-168, 2005.

[87] R. Saini, M. Cherniakov, and V. Lenive, "Direct path interference suppression in bistatic system: DTV based radar," *2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695),* pp. 309-314, 2003.

[88] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory,* vol. 28, no. 1, pp. 55-67, 1982.

[89] *ATSC Standard for Transmitter Synchronization*, Doc. A/110:2011 2011.

[90] D. Sparano, "What exactly is 8-VSB anyway?," *ARRL, Newington, CT, USA,* 1997.

[91] J. G. Proakis, *Digital communications*, 3rd ed. (McGraw-Hill series in electrical and computer engineering Communications and signal processing). New York: McGraw-Hill, 1995, pp. xxi, 928 p.

[92] C. Advanced Television Systems, "ATSC Recommended Practice: Transmission Measurement and Compliance for Digital Television," Advanced Television Systems Committee Inc.2008.

[93] J. G. Proakis and M. Salehi, *Fundamentals of communication systems*, Second Edition. ed. Boston: Pearson, 2014, pp. xx, 903 pages.

[94] C. De Boor, *A practical guide to splines*, 2nd ed. (Applied mathematical sciences, no. 27). New York: Springer, 2000.

[95] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of,* vol. 93, no. 26, pp. 429-441, 1949.

[96] M. I. Skolnik, *Radar handbook*, 3rd ed. New York: McGraw-Hill, 2008.

[97] P. M. Woodward, *Probability and information theory, with applications to radar*, 2d ed. (International series of monographs on electronics and instrumentation,, no. 3). Oxford, New York,: Pergamon Press, 1964, pp. x, 136 pages.

[98] R. N. McDonough and A. D. Whalen, *Detection of Signals in Noise*. Academic Press, Inc., 1995, p. 495.

[99] H. L. Van Trees, *Detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

VITA

Weiwei Sun earned a Bachelor degree from Northwestern Polytechnical University in Xi'an, China, with a major in Applied Physics. Dr. Sun did her graduate work at the University of Washington Department of Electrical Engineering in Seattle, and completed her PhD studies in June 2017. She is now working at Impinj Inc. in Seattle, Washington. She enjoys hiking in the Pacific Northwest with Summer, a playful but sweet dog.

Publications:

[1] 'Digital Demodulation of DTV Signals for Passive Radar Application' *Radar Conference, 2016 IEEE*, Philadelphia

[2] 'High speed computation for new digital VHF and UHF passive radar receiver', US National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM), University of Colorado, Boulder, CO. 2013

[3] 'Electro-optic thin films of organic nonlinear optic molecules aligned through vacuum deposition', Optics Express. 19(12), 11189-11195, 2011

[4] 'Real-time visualization of Karman vortex street in water flow field by using digital holography', Optics Express 27, 20342-20348, 2009

Conference Talks:

[1] 'High speed multi-antenna receiver development', CASPER Workshop, University of California, Berkeley, CA, Jun. 2014

[2] 'Further development of very high speed multi antenna receiver', CEDAR Workshop, University of Washington, Seattle, WA. Jun. 2014

[3] 'High speed computation for new digital VHF and UHF passive radar receiver', USNC-URSI NRSM, University of Colorado, Boulder, CO. Jan. 2013