12-2018

Exponent: High Speed CubeSat Radio

Dana Sorensen
Utah State University

EXPONENT: HIGH SPEED CUBESAT RADIO

by

Dana Sorensen

Capstone submitted in partial fulfillment of
the requirements for graduation with

UNIVERSITY HONORS

with a major in

Computer Engineering
in the Department of Electrical and Computer Engineering

Approved:

Capstone Mentor Dr. Jacob Gunther	**Departmental Honors Advisor** Dr. Jacob Gunther

University Honors Program Director
Dr. Kristine Miller

UTAH STATE UNIVERSITY
Logan, UT

Fall 2018

Abstract

Software Defined Radios (SDR) are limited by their usable bandwidth, which restricts how quickly they can transmit or receive data. The High Speed CubeSat Radio (HSCR) is capable of transmitting and receiving higher bandwidth signals by combining the bandwidths of several individual SDR transceivers. The HSCR performs phase correction to seamlessly stitch the bandwidths of each individual SDR into a single larger bandwidth.

By increasing the number of SDRs inside the HSCR, the bandwidth of the overall system will increase. This increase in bandwidth is limited only by the speed of the processing unit in working with received or transmitting data.

The system developed in this document is a proof of concept design. It shows that bandwidth combination is feasible and can be performed in real time.

Acknowledgements

We would like to thank Dr. Jacob Gunther for championing the idea for the project, Dr. Don Cripps for his help in planning and review, and Jolynne Berrett for reviewing and suggesting improvements to the documentation. We especially thank the Space Dynamics Lab for allowing us to use their hardware and facilities for the development of HSCR.

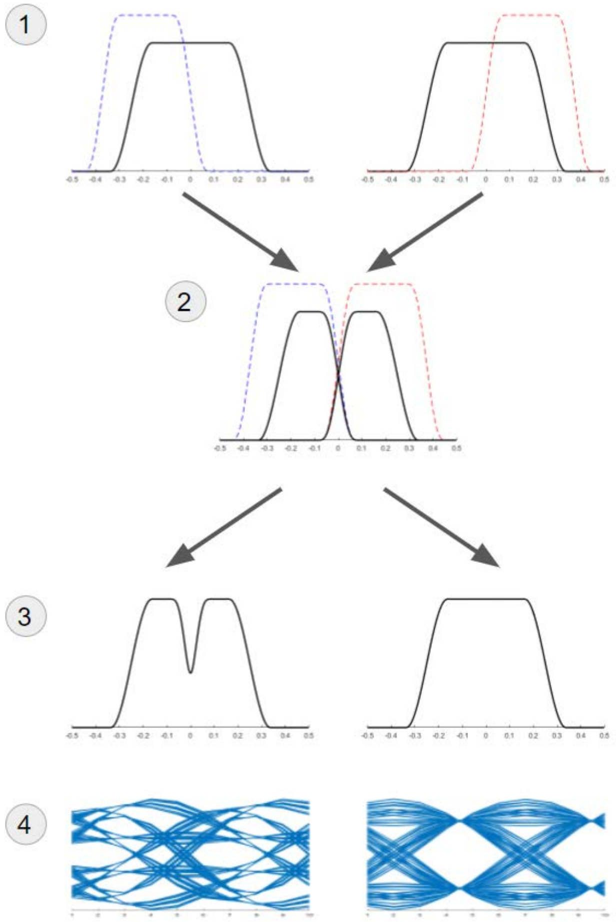
Table of Contents

Introduction

With the growing popularity of miniature satellites called CubeSats, the demand for cheap, powerful, and lightweight electrical instruments has increased. In satellite telecommunications, only a very small time window is available for data transmission between the satellite and its ground station. To more effectively use the time window provided, engineers require communication systems that can transmit at a higher data rate. This demand increases the bandwidth requirements of the system, which means radios with larger bandwidths are needed.

As the bandwidth of a radio increases, the radio's power consumption and price increase dramatically. Because CubeSats are small and limited in the amount of power they can provide, each component on the CubeSat must be small and energy efficient. This project is a proof of concept project to show that several small, affordable radios can operate as a single large radio by stitching together the bandwidths of each radio. This should effectively reduce the size, power consumption, and monetary price of the radio.

To succeed in this project, the bandwidths of two radios must be stitched together with minimal distortion. The bandwidth stitching process can be visualized by the diagram provided in Figure 1:



1) Gather data from two separate receivers tuned to different frequencies.

2) Overlap the two sources of data in preparation for combination.

3) Add signals together. Due to offsets in phase between the two signals, the signals will not combine as expected (left figure). To get valid combination, the phase offsets must be corrected (right figure).

4) When phase is not corrected, combining the two signals produces a distorted eye diagram (left figure). When phase is corrected, the eye diagram shows no distortion (right figure).

Figure 1: Signal combining concept.

Materials

This section documents the components used in the design of the project. Decision matrices have not be provided, because each components used in this project was selected by the people funding the project. This section will simply provide information on each of the components.

USRP B205mini-i

The B205mini-i is a software defined radio manufactured by Ettus Research. This component will be used as a transmitter in our project design. It can be controlled using premade signal blocks in GNU Radio. This radio is capable of transmitting at frequencies between 70 MHz and 6 GHz using up to 56 MHz of bandwidth. For testing purposes, our system will be transmitting at 2.4 GHz. This component is currently priced at $910.00.

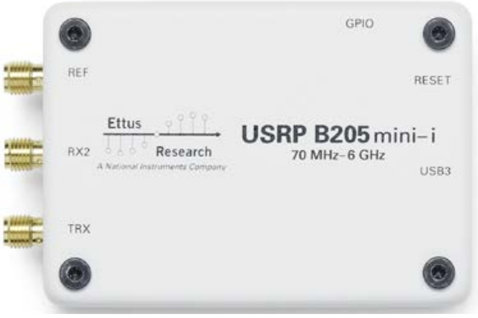


Figure 2: USRP B205mini-i.

AD-FMCOMMS5-EBZ

The AD-FMCOMMS5-EBZ is a development board manufactured by Analog Devices. The board features two AD9361 transceivers that share common oscillators. The common oscillators are designed to create a phase coherent system. This board is used as the receiver in our system. In interfaces directly with our FPGA board.

The AD9361 transceivers are capable of receiving signals between 70 MHz and 6 GHz. The bandwidth of the receivers is configurable between 200 kHz and 50 MHz. As stated above, these receivers will be tuned to receive data at 2.4 GHz. This component is currently priced at $1,125.00.

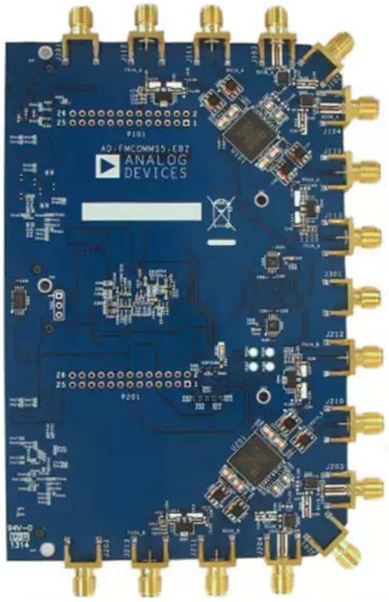


Figure 3: AD_FMCOMMS5-EBZ.

Xilinx Zynq-7000 All Programmable SoC ZC706

The Xilinx Zynq-7000 is a FPGA development board that can directly interface with the FMCOMMS5 board through a FMC parallel connection. This board is programmable using the Vivado design suite. Additionally, this board comes with a built in microprocessor running Linux. This allows us to run GNU Radio on the FPGA development board directly. This component is currently priced at $2,495.00.

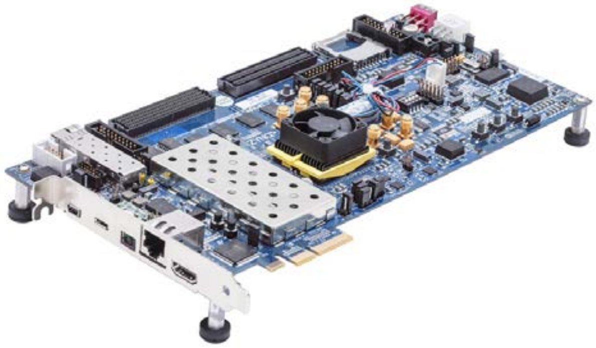


Figure 4:Xilinx Zync-7000 FPGA Development Board.

Methods

This section will discuss the system and functional design of the HSCR. The system design shows how each of the components from the Materials section are put together to form the system. The functional design includes the information on the algorithms used to accomplish the project goals.

System Design

A block diagram is provided below in Figure 5 that shows the system structure. The system can be thought of as two separate pieces: the transmitting components and the receiving components.

The transmitting portion of the system is made up of the B205mini-i transceiver and a host PC. The Host PC runs GNU Radio and sends data to the B205 transceiver via a USB connection. The receiving portion of the system is made up of the FMCOMMS5 development board and the Zynq-7000 development board. The FMCOMMS5 and Zynq boards interface with each other through a FMC parallel connection. Because the Zynq board is acting as the host PC for the receiver portion, a computer monitor, keyboard, and mouse are connected to the Zynq board.

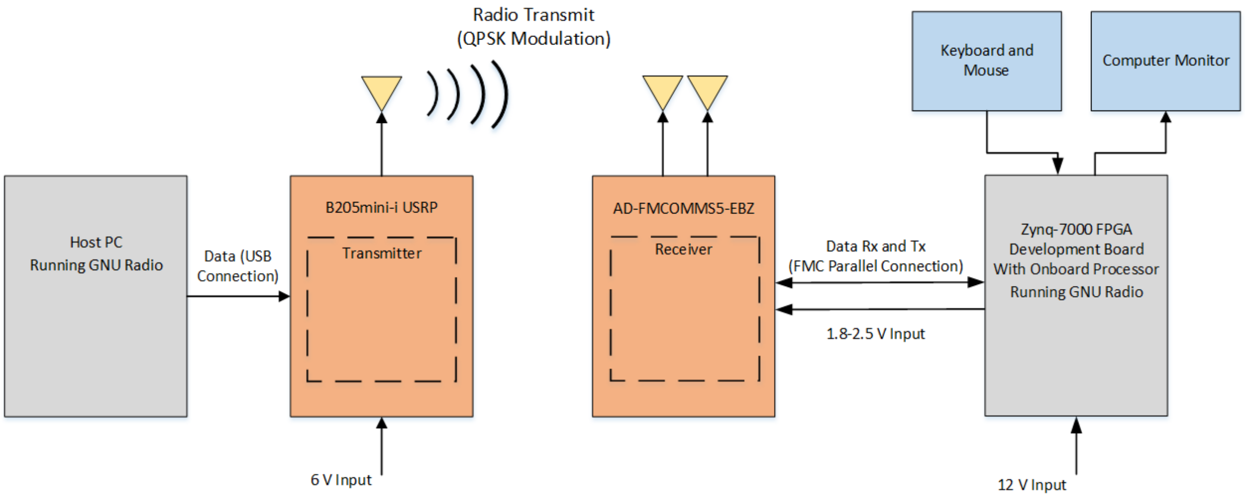


Figure 5:System block diagram.

Functional Design

On a high level, the system is broken into three sections. First, a signal is transmitted from a USRP b205 radio. The signal is then received by the HSCR on the FMCOMMS5 receivers where the signal bandwidths are then combined. Finally, the combined signal goes through phase, timing, and gain correction to recover the transmitted signal.

The block diagram shown in Figure 6 outlines the high level structure of the project. Each section of the high level block diagram will be discussed in further detail within this section.

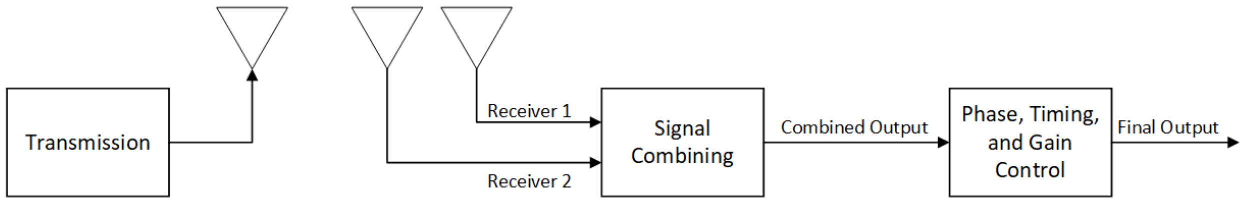
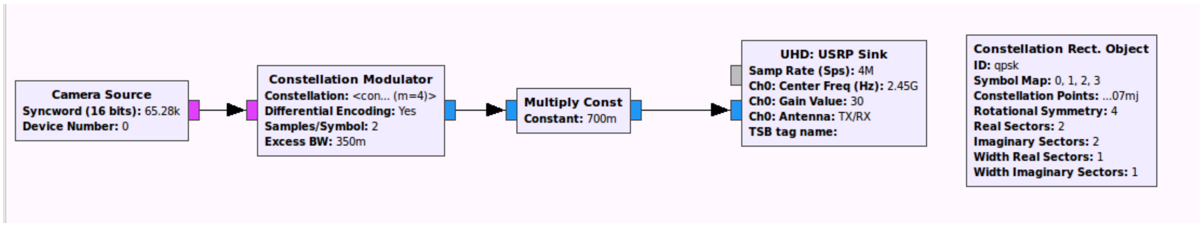


Figure 6: High-level functional block diagram.

Transmission

As stated in previous sections, the transmitting portion of the system will be handled by the B205mini-i. The GNU Radio flowgraph controlling transmission is shown in Figure 7 below. For this particular flowgraph, an image is captured using OpenCV inside of a custom GNU Radio block called "Camera Source." The captured image is then converted from bits to symbols and is transmitted using a USRP block designed to interface with the B205mini-i.



Camera Source
Syncword (16 bits): 65.28k
Device Number: 0

Constellation Modulator
Constellation: <con... (m=4)>
Differential Encoding: Yes
Samples/Symbol: 2
Excess BW: 350m

Multiply Const
Constant: 700m

UHD: USRP Sink
Samp Rate (Sps): 4M
Ch0: Center Freq (Hz): 2.45G
Ch0: Gain Value: 30
Ch0: Antenna: TX/RX
TSB tag name:

Constellation Rect. Object
ID: qpsk
Symbol Map: 0, 1, 2, 3
Constellation Points: ...07mj
Rotational Symmetry: 4
Real Sectors: 2
Imaginary Sectors: 2
Width Real Sectors: 1
Width Imaginary Sectors: 1

Figure 7: Transmission flowgraph in GNU Radio.

Signal Combination

Once the transmitted signal is received by the two receivers on the FMCOMMS5 board, the signals must go through a signal stitching process. This process is split into several smaller pieces. A block diagram of the signal combining process is shown below in Figure 8. This process is implemented inside the FPGA.

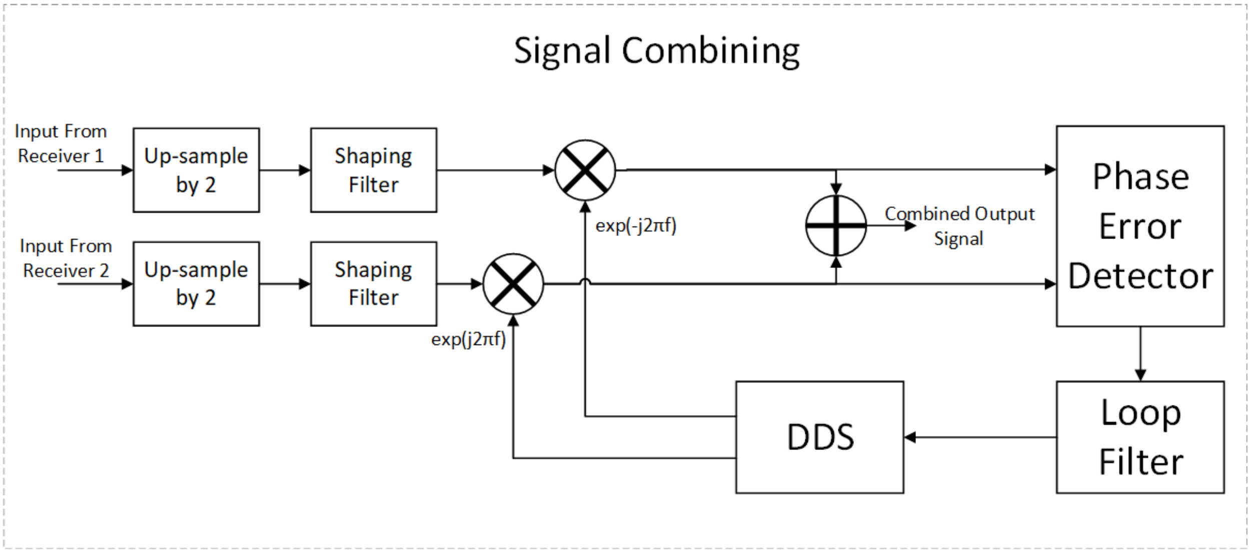


Figure 8: Signal combination flow chart.

For successful signal combination, the two signals must add together in the frequency domain without any magnitude or phase distortion. Any distortion in either the magnitude or phase responses will corrupt the digital data and invalidate the combining process.

The first step in combining the signals is to shape the bandwidths using a specially designed Nyquist filter. The Nyquist filter was designed using an iterative method that transforms a normal low-pass filter into a Nyquist filter. Having a Nyquist filter allows the two received signals to be added together without any magnitude distortion. To correct for phase distortion, the signals must go through a phase error control loop. This loop consists of a multiplication by a complex exponential, a phase error detector, a loop filter, and a DDS that controls the complex exponential.

The complex exponential shifts one received signal up in frequency and shifts the other received signal down in frequency. These frequency shifts place each of the signals in the proper places in order to be added together. After the complex exponential, the two signals enter the phase error detector.

A diagram of the phase error detector is shown below in Figure 9. As the shifted signals enter the phase error detector, the signals pass through a low-pass filter. This filter isolates the part of the two signals that overlap with each other. Using the overlapping sections of the signals, an error signal is created. The error signal is then passed through a low pass filter in preparation for the loop filter.

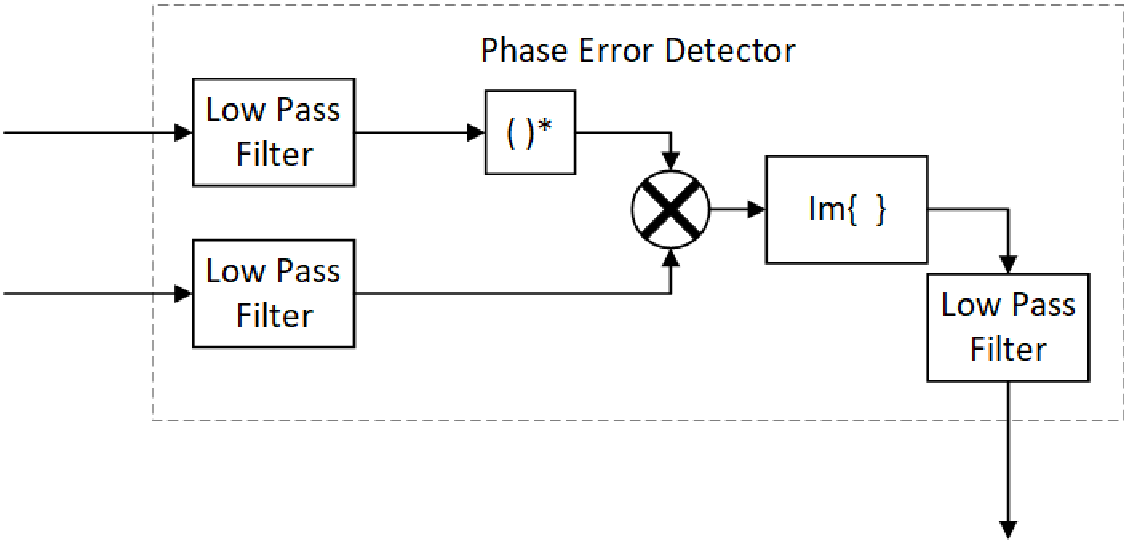
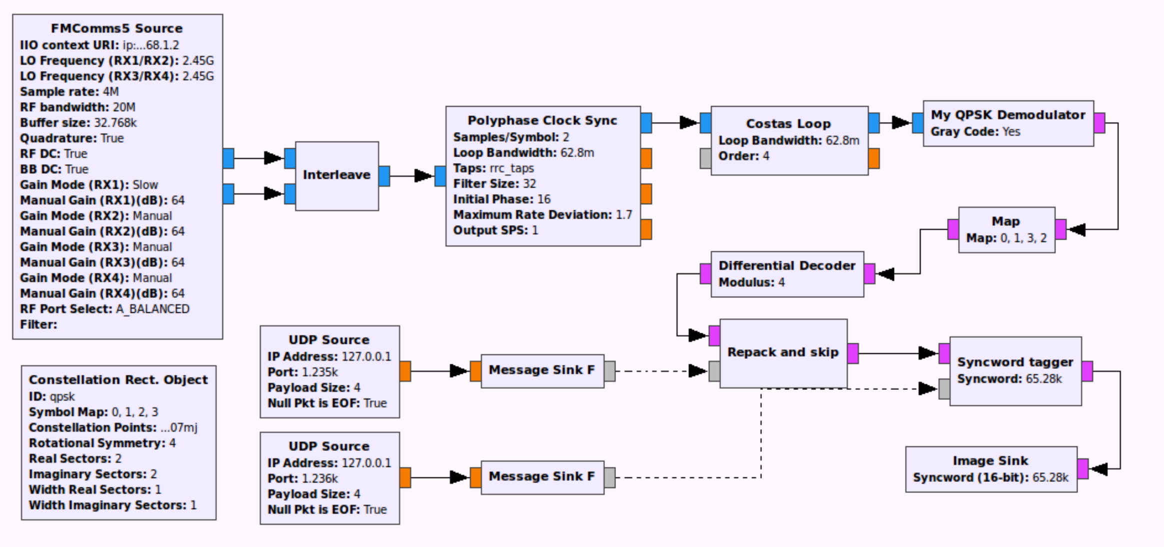


Figure 9: Phase error detector block diagram.

The loop filter is a standard proportional plus integral loop filter. This loop filter produced a filtered error signal that is used to control the DDS. The DDS then makes subtle changes in the complex exponential terms that shift the two received signals to positions that produce no phase distortions when added together.

Phase, Timing, and Gain Control

After the signals from the two receivers have been combined, they are now ready for phase, timing, and gain, corrections. GNU Radio provides several blocks that make these corrections very easy and quick to design. Figure 10 is provided below which shows the GNU Radio flowgraph implementing the needed corrections.



FMComms5 Source
IIO context URI: ip:...68.1.2
LO Frequency (RX1/RX2): 2.45G
LO Frequency (RX3/RX4): 2.45G
Sample rate: 4M
RF bandwidth: 20M
Buffer size: 32.768k
Quadrature: True
RF DC: True
BB DC: True
Gain Mode (RX1): Slow
Manual Gain (RX1)(dB): 64
Gain Mode (RX2): Manual
Manual Gain (RX2)(dB): 64
Gain Mode (RX3): Manual
Manual Gain (RX3)(dB): 64
Gain Mode (RX4): Manual
Manual Gain (RX4)(dB): 64
RF Port Select: A_BALANCED
Filter:

Polyphase Clock Sync
Samples/Symbol: 2
Loop Bandwidth: 62.8m
Taps: rrc_taps
Filter Size: 32
Initial Phase: 16
Maximum Rate Deviation: 1.7
Output SPS: 1

Costas Loop
Loop Bandwidth: 62.8m
Order: 4

My QPSK Demodulator
Gray Code: Yes

Map
Map: 0, 1, 3, 2

Differential Decoder
Modulus: 4

Constellation Rect. Object
ID: qpsk
Symbol Map: 0, 1, 2, 3
Constellation Points: ...07mj
Rotational Symmetry: 4
Real Sectors: 2
Imaginary Sectors: 2
Width Real Sectors: 1
Width Imaginary Sectors: 1

UDP Source
IP Address: 127.0.0.1
Port: 1.235k
Payload Size: 4
Null Pkt is EOF: True

UDP Source
IP Address: 127.0.0.1
Port: 1.236k
Payload Size: 4
Null Pkt is EOF: True

Syncword tagger
Syncword: 65.28k

Image Sink
Syncword (16-bit): 65.28k

Figure 10: Receiver flowgraph in GNU Radio.

Results

Results for this project came in two phases. First, we created simulations to verify that the system would work mathematically. These simulations were run in GNU Radio and MATLAB. After the simulations came out positive, we created the system in actual hardware. The system was implemented on an FPGA and results were gathered from the FPGA visualization tools preloaded on the Xilinx development board. This section will cover the results obtained from both simulation and hardware.

Simulated Results

This section will detail the simulations run in MATLAB. These simulations were performed on data that was logged from actual data on the FMCOMMS5 development board. These simulations were used to design the needed Nyquist filters and to fine tune the signal combination algorithm. The results of the MATLAB simulation are shown below in Figure 11.

The figure includes three separate plots. The top plot shows the output of the filtered phase error signal. This error signal is used to adjust the DDS in the phase error control loop shown in Figure 8. Notice that the error signal settles around an average value after a few milliseconds. This shows that the phase error loop locked on to a corrected phase value.

The middle plot in Figure 11 shows and eye diagram after the phase error control loop has locked onto the phase error. Notice that there are clear crossing points where the signals could be sampled. These sampling points occur at positions 4 and 8 in the eye diagram.

The bottom plot in Figure 11 displays two curves. The blue curve is the magnitude spectrum of the combined signal after phase error correction. The red curve shows the magnitude spectrum of the combined signal without any phase error correction. Notice the red curve has a large dip in the center of the spectrum. This dip is distortion caused by phase offsets. The presence of the dip destroys digital data and renders the signal unusable. In contrast, the blue curve shows no dip in the spectrum. This means the signals combined successfully and can be used to recover digital data.

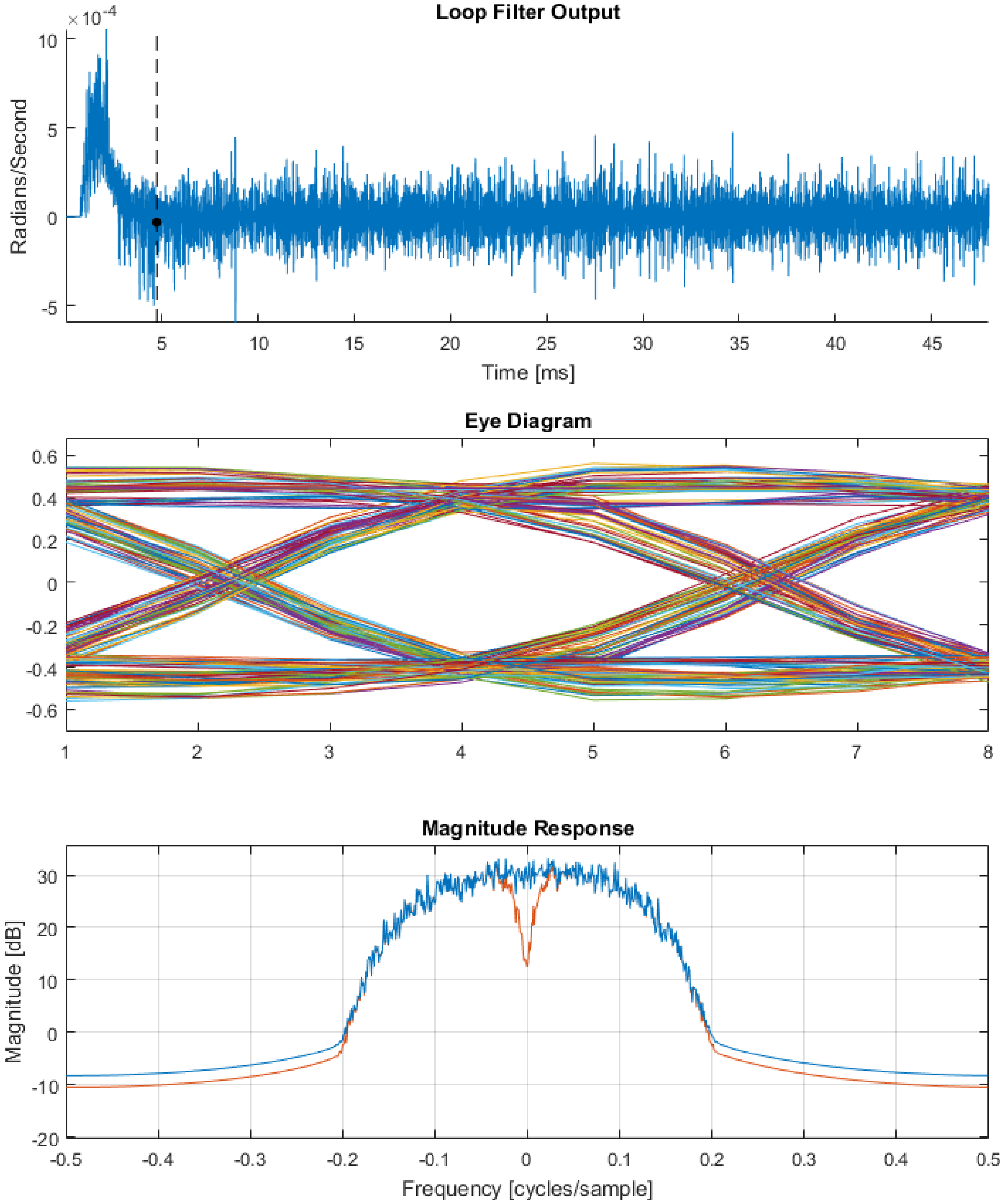


Figure 11: Simulation of phase error correction.

Actual Results

After verifying that the signal combining algorithms were correct, the design was implemented on the Zync-7000 FPGA development board. The Zynq board provides a program called iioscope, which can be used to gather information on the data passing through the FPGA. The results of the FPGA implementation are shown below in Figure 12.

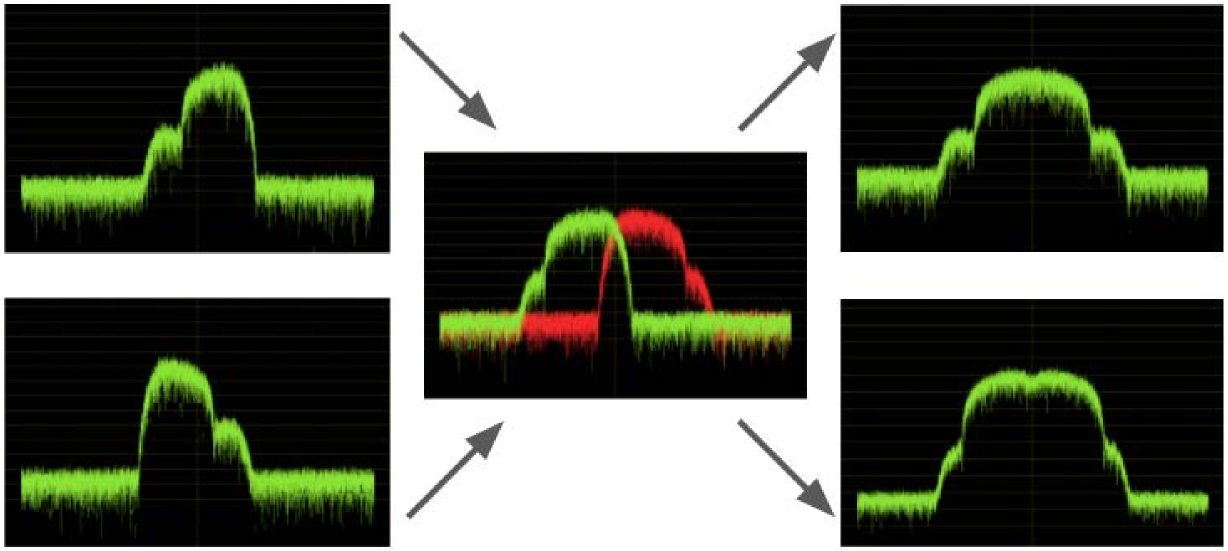


Figure 12: Actual signal results gathered from the HSCR.

In Figure 12, two signals were captured by the FMCOMMS5 board and shaped using our custom Nyquist filter. The two received signals are shifted and added in preparation for signal combination. On the right side of Figure 12, two graphs are given which show the two possible outcomes of signal combination. Of the two graphs, the top graph is the result of combining the signals with the phase error correction loop. The bottom graph is made without phase correction and shows how the signal distorts near the middle of the spectrum.

Note that the actual results reflect the simulation results. With phase correction, the spectrum contains no dip in magnitude. Without phase correction, the spectrum contains a dip. The dip in the actual results appears much smaller than the simulated results. There are two reasons for this difference. First, the simulated results are plotted on a linear scale. The actual results are plotted on a logarithmic scale. Second, the dip in this case is smaller, but even such a small dip causes enough distortion in the signal to make digital data unusable.

Because we see that the phase error correction produces a spectrum without any distortion, we know that the HSCR is performing as expected.

Testing and Verification

This section documents the testing phase of the project. Testing and verification is broken into three sections: interface requirements, functional requirements, and support requirements. Comments have been provided for each stage of testing

Interface Requirements

1. The system shall be capable of receiving signals transmitted by existing consumer radios.
2. The system shall transmit signals that can be received by existing consumer radios.
3. The system shall be able to receive and transmit any signal regardless of modulation scheme.
4. The system shall be capable of transmitting and receiving at any user specified frequency inside the tuning range of the SDRs.

Requirement	Status	Comment
#1	Met	A signal was transmitted from a Ettus USRP B205mini-i radio and was received successfully using the HSCR.
#2	Incomplete	For prototyping and proof of concept purposes, our system was set up for receiving signals only. Very few modifications would be needed to produce a system that could transmit signals as well.
#3	Incomplete	The HSCR used QPSK as a primary proof of concept modulation scheme. The system was built to adapt to other modulation schemes, but has not been verified.
#4	Met	The tuning frequency of the radio was varied to verify functionality. Total bandwidth is also configurable, but requires modification of the FPGA design. Configuring bandwidth size will be an area of further research.

Functional Requirements

1. The system shall consist of more than one SDR.
2. The system shall be capable of both receiving and transmitting RF signals.
3. The bandwidth of the transceiver shall be greater than that of the individual SDRs.
4. The transceiver shall consume less power than a comparable traditional transceiver.
5. The transceiver shall be clock and phase coherent.

Requirement	Status	Comment
#1	Met	The HSCR uses two radios and is scalable.
#2	Incomplete	See comment for interfacing requirement #2
#3	Met	The spectra of the individual radios was compared to the spectrum of the combined signal.
#4	Incomplete	Time did not permit verification of this requirement. Additional time and research is needed.
#5	Met	A phase error detector has been implemented to correct phase decoherence. Analysis of received data has shown phase coherence.

Support Requirements

1. The system shall survive temperatures between -30 and 60 degrees Celsius. (Note: for prototyping purposes, the system shall survive temperatures between 0 and 40 degrees Celsius.)

Requirement	Status	Comment
#1	Met	The prototyping requirement was met by the specifications given for each individual component in the system.

Discussion

Summary

According to both the simulation data and the actual data, it is possible to receive separate signals on two different receivers and combine their bandwidths to create a single combined signal. This system proves that signal combination works for receiving signals. Minor modifications would show that the system can also transmit two separate signals that a different receiver could see as a single signal.

Bit Errors and SNR

To further analyze the HSCR system, a bit error rate plot was generated. This plot is shown in Figure 13. This figure is the result of counting bit errors in the presence of noise. Different signal to noise ratios were used to generate the points of the plot. Notice that the HSCR actually performs better than a comparable radio. This is due to the redundancy of data in the overlapping region of the combined signal. With two sources of the same data, noise can be averaged out and improve the overall signal.

After testing, we noticed that the error rate continued to improve as the overlapping area increased. We also noticed; however, that increasing the overlapping area had the effect of cutting off data on outside edges of the signal. There is a tradeoff here between signal fidelity and error rate improvement.

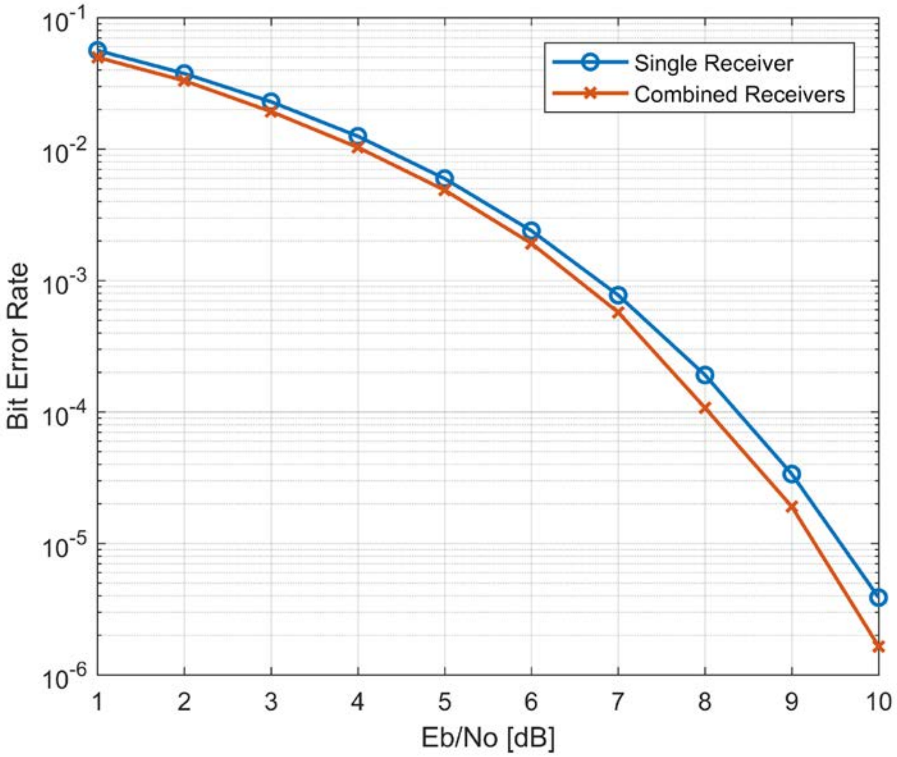


Figure 13: Bit error rate at varying SNR.

Further Research

This project could grow in many different directions. One area of further research would be to combine the bandwidths of more than two receivers. The system also needs to be tested transmitting signals as well as receiving signals.

Due to time constraints, we did not perform a power analysis. One of the major objectives of the project was to create a radio that consumes less power than a commercially available radio of similar performance. Further research would tell us whether or not this system is more power efficient.

Further work can be done in packaging the system. Currently, there are several things to keep track of when setting up the system. Given some time, the system could be packaged into an easy to use GNU Radio block that would handle each of the technicalities that must currently be handled by a user.

For ease of demonstration, we used GNU Radio to implement phase, timing, and gain control. This method was easy to implement, but wasn’t as fast as we had hoped. Further research would let us implement the phase, timing, and gain controls on the FPGA. This would allow for significant speed boosts in data transmission.

Potential Applications

The primary application of this system is as a satellite telecommunications system. This system can be applied in many other situations as well. This project can be used in any system that requires a high speed communication system. Example systems could be military/civilian communication networks, space communication, aircraft and watercraft uplinks, mobile internet, etc.

Conclusion

By algorithmically correcting for phase offsets, the HSCR was able to successfully combine the bandwidths of two separate receivers. With very few changes to the system, the phase correction technique can apply to systems with more than two radios to further increase the system bandwidth.

One of the main goals of the project was to create a system that consumes less power than a typical system of similar capabilities. Given the time frame of the project, further study will be required to determine the power consumption of the system. Further work is also needed to create a more user friendly interface.

The results of this project show promise of influencing the current CubeSat telecommunications industry.

Reflection

This capstone project was a great learning experience for me. I enjoyed the opportunity to start from scratch with a goal in mind, choose the approach, pick out the hardware, and implement everything needed to meet the goal. The entire process supplemented my classwork and previous experience, and I was able to get to know my mentor better as we worked through several roadblocks.

The project coincided well with the theoretical knowledge I have received from my classes at USU. For example, while working on this project I took a digital communications class where I learned about modulation schemes, analysis methods, and system characterizations for digital communication systems. I used nearly every concept from that class in the development of HSCR. I also took a digital design class focused on FPGA development, and I was able to use the same tools and ideas to develop the real-time implementation for the project. One semester ago I didn’t know anything about these concepts, but now I have both the academic, theoretical knowledge from my classes and a practical, working knowledge from this project.

I was able to develop a better working relationship with my mentor and department head, Dr. Jacob Gunther. I met with Dr. Gunther weekly to discuss our progress and refine our strategies. He gave invaluable advice and pointed me us the right direction each week. Without his help, the project would have taken much longer and would not have produced such great results. As a result of working closely with Dr. Gunther, I have been able to work with him as my Major Professor to choose and start my Master’s Thesis.

The thing I enjoyed most about the project was the critical thinking required to overcome the obstacles we ran into. I was able to apply the theory in order to understand the behavior of the system, fix problems, and improve performance. These experiences helped me gain better intuition for the things I already knew. For example, it was satisfying to break the sub-filter incoherency problem into mathematical concepts to explain the behavior, then use that mathematical model to propose and implement a solution. I also enjoyed creating figures and diagrams to explain the problem and solution in an easy to understand and approachable way.

The greatest triumph came when we finished the implementation and began to characterize it. Having just learned about the tools and methods most commonly used to classify communication systems in my classes, I generated a bit error rate plot (Figure 13 in the document above). Both Dr. Gunther and I were initially very surprised to see the system performing better than the theoretical best for a single receiver. At first we assumed a mistake had been made in the characterization, but after further analysis, we determined that any implementation loss in the system was overcome by the reduced signal-to-noise ratio in the overlapping region. In other words, our two-radio combination for HSCR can receive signals more clearly than a traditional single radio!

The greatest thing that prepared me for this Capstone Project was the research I have been involved in. I started doing research with ECE faculty my first semester at USU, and I’ve had the opportunity to work with three of my professors on various research projects. Those projects gave me the experience I needed to successfully complete this project.

Author Bio

Dana Sorensen is a student at Utah State University. He is currently pursuing a Bachelor’s Degree in Computer Engineering with Computer Science and Math minors. He is a member of the USU Honors Program, the USU IEEE student branch, the Tau Beta Pi engineering honor society, and is an Undergraduate Research Fellow. During his time at USU, he has had the opportunity to work with three ECE faculty members on various research projects and complete two technical internships. He helped found the Autonomous Vehicle Club and is heavily involved with the USU Mars Rover Team. Dana currently works at the Space Dynamics Lab in Logan, Utah, and plans on pursing a Master’s degree in Electrical Engineering at Utah State University.