STEVENS INSTITUTE OF TECHNOLOGY

SDR Amateur Repeater

Project Report

Erik Thompson, Matt Schurmann, Scott Curtis, Jon Pirog

We pledge our honor that we have abided by the Stevens Honor System

5/4/2011

This document provides an overview of the functional, and component design as well as a discussion of design tradeoffs involved in the development of software defined radio based repeater for amateur radio use. The design was developed for a sixth course in engineering design at Stevens Institute of Technology, during the Spring of 2011 and is intended to be implemented as a final project for graduation from the bachelors program of its Electrical and Computer Engineering department.

CONTENTS

Team Member Responsibilities 3
Section 1: Introduction 4
Section 2: Technical Information 5
  Section 2.1: Functional Description of the Design and its Components 5
    System-Level Overview 5
    RF (Radio Frequency) Subsystem 5
    IF (Intermediate Frequency) Subsystem 7
    Baseband Processing Subsystem 8
    Baseband Functionality 8
  Section 2.2: Technical Description of the Design and its Components 9
    RF/IF Front-End 9
    Baseband Processing 12
    Associated Trade-Offs 13
    Baseband Design Alternatives 15
    Most Viable Alternatives 16
    Possible Alternative: SDR Kit + GNU Radio 16
  Section 2.3 Mathematical & Other Principles 16
    Analog Modulation 16
    Mathematical Basis for Frequency Modulation 17
    Sampling Theory 17
    DSP: Finite Impulse Response Filtering 17
    DSP: The Fast Fourier Transform 18
    DSP: Direct Waveform Synthesis 18
  Section 2.4 Performance Expectations 18
    Noise Considerations 18
    Spurious Responses 19
    Selectivity 19
    Image Rejection 19
    Transmit Power 20
Section 3: Critical Evaluation 21
  Section 3.1: The “Good” 21
  Section 3.2: The “Scary” 21
  Section 3.3: The “Fun” 21
  Section 3.4: Funding of Project 21
Section 4: Conclusions 22
Section 5: Appendices 23
  Appendix 1: Team Member Vita 23
  Appendix 2: Links to Datasheets 24
  Appendix 3: References 24

LIST OF TABLES AND FIGURES

Figure 1: A Repeater 4

Figure 2: Functional RF Frontend Architecture 6

Figure 3: Functional IF Subsystem 7

Figure 4: Functional Software design 8

Figure 5: Receiver Schematic 11

Figure 6: Transmitter Schematic 11

Figure 7: Altera cyclone fpga demo board 12

Figure 8: FPGA/GPP Baseband 15

Figure 9: GPP/DSP Baseband 15

Figure 10: Analog Modulation 16

Figure 11: Tapped Delay line filter 17

Figure 12: Direct Waveform Synthesis 18

Figure 13: Illustration of the Third Order Intercept 19

Figure 14: Image Frequencies 19

TEAM MEMBER RESPONSIBILITIES

Erik Thompson

- Section 2.4
- Section 2.3

Matt Schurmann

- Section 1
- Schematics
- Section 4
- Appendices
- Report Organization

Scott Curtis

- Section 2.1
- Section 3.1, 3.2

Jon Pirog

- Section 2.3
- Section 3.3, 3.4

Section 1: Introduction

Since the dawn of radio, there have been radio hobbyists. These early hobbyists evolved into today's amateur radio operators. They are licensed for noncommercial use of the radio spectrum, and are involved in recreation, experimentation, and emergency assistance. Amateur radio operators, also known as "hams," use various modes of transmission to communicate, including morse code, analog and frequency modulation (AM and FM).

In the VHF and UHF bands, FM is used for most contacts, which are often made using repeaters. Repeaters serve to extend the range of a user by receiving a weak signal on one frequency and then transmitting a more powerful signal on another. They are often placed at a higher altitude when compared to the area they service, such as on a tall building or on the summit of a tall mountain, which serves to further extend the range of weak signals. When better reception of an amateur radio signal is needed, and it is impossible to simply transmit it with more power, a repeater is used. A repeater sits between the two users, receiving each users signal at the edge of their range, and echoing it with enough power to reach the other user.

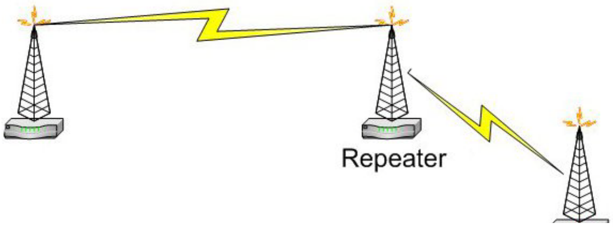

FIGURE 1: A REPEATER

Today, most repeaters are implemented with fixed-function RF hardware. They receive and transmit in a single (or few) of the amateur radio bands, using one modulation scheme, and are quite large, and expensive to maintain. In a world of ever-changing spectrum allocations, and increased usage, they offer no cheap method of re-configuration, and require separate hardware for each function they perform.

These fundamental limitations on fixed-function hardware radios can be addressed with the use of software-defined radio (SDR). Software defined radio is a radio architecture paradigm that seeks to implement as much of a radio's functionality as possible through programmable, re-configurable baseband digital signal processing. Only recently have the computing power of baseband hardware, and the flexibility of RF hardware been able to support the requirements that software defined radio places on them.

The objective of the project described herein is to implement a prototype for an amateur radio repeater, using software defined radio techniques, a flexible RF frontend, and re-configurable baseband hardware that will be able to usably function in any of the currently licensed amateur radio bands in the United States, meeting all FCC requirements for commercial and personal use. The design will be implemented as a final senior-level engineering design project during the academic year of Fall 2011 – Spring 2012, at Stevens Institute of Technology, in Hoboken, New Jersey.

As software defined radio is still somewhat of an emerging technology, much research into component specification, functionality, and design trade-offs was needed to develop the following report, and the component and design specifications it entails.

The following sections of the report detail technical background information, needed functionality, component specification considerations, and the mathematical basis for creating the hardware necessary to implement the objective. Following technical information is a critical discussion of the project as a practical senior design project, conclusions, and references to the many papers and sources researched for the purposes of the project.

SECTION 2: TECHNICAL INFORMATION

This section will provide an overview of the technical research, implementation issues, and design efforts made toward the eventual implementation of the project as a senior-level design project in the Fall 2011 – Spring 2012 academic year.

SECTION 2.1: FUNCTIONAL DESCRIPTION OF THE DESIGN AND ITS COMPONENTS

Having done some extensive research into the requirements and practical considerations of an SDR-based repeater - it is important to have an idea of what *functions* the different components of the system will need to have, rather than moving directly into hardware/software implementation. This subsection is meant to explain in some detail, the functional block diagrams for the system.

SYSTEM-LEVEL OVERVIEW

The system is defined as a software-defined amateur radio repeater. Implementation of both hardware and software is required for the overall system to function to specifications and design parameters. The hardware sub-systems are in charge of the physical reception and transmission of any relevant signals. In addition, they are responsible for converting the signal into data that can be utilized by the baseband processing portion of the system. The hardware subsystem will modify the baseband processing output in order to prepare the signal for transmission.

The baseband processing sub-system is responsible for the user input and is the key component to the SDR system. The sub-system is responsible for the modulation and demodulation of the baseband signal in addition to managing tone access. The user input is responsible for manipulating the tone frequency; transmit power, transceiver frequency, and transceiver bandwidth. These parameters are processed through software which limits the amount of hardware exchange in order to change the input and output parameters of the system. All of the sub-systems congeal to create a repeater with a software-defined foundation with auxiliary hardware support.

RF (RADIO FREQUENCY) SUBSYSTEM

Typically the most hardware-intensive portion of an SDR, the RF section is the only section of the SDR that must use mostly passive (non-software controllable) components.

As will be seen in the following sections, the RF frontend will present the most implementation challenges of the different subsystems employed. Components will need to trade gain, bandwidth, re-configurability, and power consumption.

Basic responsibilities of the RF section are as follows:

- Receiving and amplifying the input signal
- Noise Rejection
- Conversion to and from IF frequency to RF
- Amplifying DAC output, and transmitting signal over a channel

Consider the basic RF receiver front-end architecture for an SDR and the functionality for each component listed on the right.

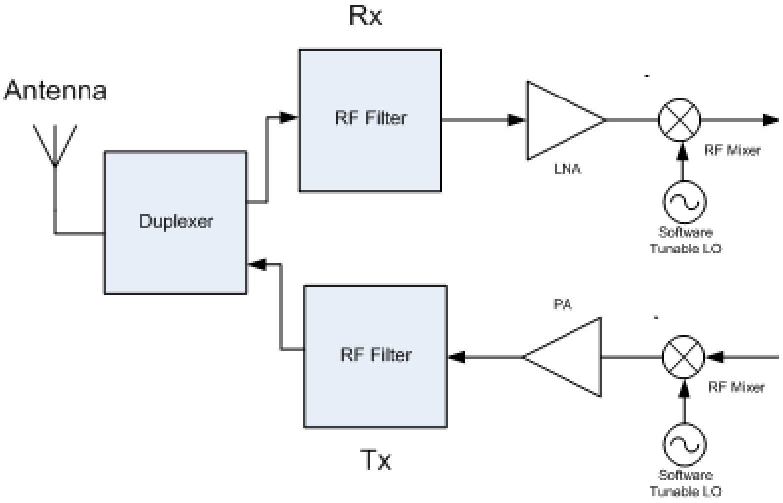

FIGURE 2: FUNCTIONAL RF FRONTEND ARCHITECTURE

Antenna: Antennas transmit and receive RF (radio frequencies) with sufficient power and sensitivity on many frequencies, time frames, and spatial areas. An SDR antenna system must provide sufficient diversity for transmission and receive a wide range of frequencies and in many different environments. Much design time needs to be dedicated to implementation of duplexing and wideband antenna/antenna array system development. The antenna is solely connected to a duplexer in this SDR implementation.

Duplexer: Duplexers allow bi-directional (duplex) communication over a single channel. A duplexer must be designed for operation in the frequency band used by the receiver and transmitter, and must be capable of handling the output power of the transmitter. A duplexer must provide adequate rejection of transmitter noise occurring at the receive frequency, and must be designed to operate at, or less than, the frequency separation between the transmitter and receiver. A duplexer must provide sufficient isolation to prevent receiver desensitization. The duplexer will be communicating two separate signals to and from the RF filters to the antenna as shown in the diagram.

RF Filter: Radio frequency (RF) filters represent a class of electronic filter, designed to operate on signals in the megahertz to gigahertz frequency ranges. This frequency range is the range used by most broadcast radio, television, wireless communication (cellular phones, Wi-Fi, etc...), and thus most RF and microwave devices will include some kind of filtering on the signals transmitted or received. Such filters are commonly used as building blocks for duplexers and diplexers to combine or separate multiple frequency bands. In the scope of this project, a band-pass filter is a type of RF filter that passes frequencies within a certain range and rejects (attenuates) frequencies outside that range.

LNA and PA: Very important to the system performance, is the addition of a low noise amplifier (LNA) and power amplifier (PA). On both the receiver and transmit sides, power amplification will be required – as received signals are generally low-power due to path losses, and for power concerns, is impractical/unnecessary to operate the entire system at enough power to transmit over a wireless channel. Most importantly, the LNA selected will need to be very linear, and have a considerably low noise figure. Amplification may generally be completed in multiple stages, especially at the transmit side, where linearity, and gain are equally important to system performance.

RF Mixer and Software LO: The functionality of the RF mixer is to mix the amplified analog signal down to a lower frequency before it can be sampled practically. Very high speed ADC modules will be prohibitively expensive in the RF regime, so a variable mixer is used to convert the signal to a more manageable frequency for ADC/DAC. The local oscillator (LO) used for mixing, can be controlled with software.

IF (Intermediate Frequency) Subsystem

The IF stage of an SDR system is where the RF signal that received/transmitted must make the transition from the digital to the analog realm on the transmit side, and vice versa on the receiver side. The SDR paradigm dictates that as much of the radio's function as possible should be implemented in software – necessitating digitization, even though the actual end-user functionality of the radio is analog (RF).

The IF section is notable for the fact that it places quite a number of practical constraints, and necessitates a number of trade-offs for an SDR system design. Analog and digital conversion technologies are not yet sufficiently fast, cheap, and accurate enough to adequately sample RF communications in a practical manner. These constraints require the use of a mixer in the RF subsection to decrease the operating frequency of a signal prior to analog/digital conversion, so that some of these requirements can be relaxed a bit.

In addition to the use of a mixer to relax ADC/DAC constraints, a digital conversion stage is also necessary to synchronize the signal to the operating frequency of the baseband section. See the figure below.

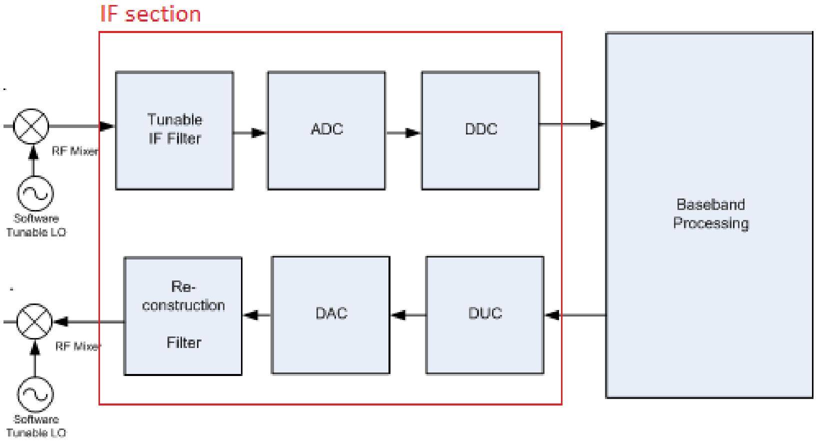

FIGURE 3: FUNCTIONAL IF SUBSYSTEM

In general, ADC architecture will be critical to receiver performance. Although a mixer was used in the RF section to loosen some of the ADC speed requirements, MHz range is still a considerably high frequency to quickly convert an analog signal to a baseband signal with many bits of precision. Traditionally, a flash ADC architecture would be used for speed, but since they generally provide few bits of precision, it is also advised to use a pipeline ADC, which basically uses a series of low-resolution flash ADC components to implement a type of binary search on the signal. Similarly, a modified pipeline architecture can be employed for the DAC stage on the transmit side as well.

A baseband processor unit and an IF signal are not fully compatible, however. It is therefore necessary to do a digital form of mixing, conversion – up-conversion before the DAC for transmission, and down-conversion before the baseband processor at the receiver. This digital conversion step will convert the signal from an IF signal to a baseband signal on the receiver side, and prepare it for DAC, and RF mixing on the transmit side.

Generally, digital up/down-conversion is implemented using a field programmable gate array (FPGA), which although user configurable is not entirely ideal, as it will generally not allow the system to be configured by software in real-time. Similarly, some SDR designers also choose to use an application-specific integrated circuit (ASIC) to perform digital conversion. For real-time configurability, it is recommended that a digital signal processor (DSP) be used to perform conversion in the IF stage of the SDR architecture, because generally, baseband software will be able to manipulate DSP operation based on the requirements it sees, which could change in real-time.

Baseband Processing Subsystem

While it is very important that significant performance be achieved in the previous two stages, it is the baseband section of the system that will require the most attention from a designer – primarily, because it is the baseband processor that will be responsible for the main functions of the system:

- Encoding/Decoding
- Interleaving
- Modulation/Demodulation
- Baseband DSP, filtering etc…
- Error correction
- Signal Correlation
- Data Link layer interfacing
- On-the-fly configuration of RF/IF hardware when needed

Typically, implementations of a baseband section will either use an FPGA, DSP, or a dedicated microcontroller and related digital circuitry (memory, power, etc…). Probably the most versatile option for baseband processing is the microcontroller – as it will generally be the most powerful, easy to program/interface with, and compile code for. However, there is a tradeoff involved with using a microcontroller-based architecture, as this type of system will generally require a good deal of power.

Programming for the system will need to be done in a low-level language: C/Assembly for a microcontroller, or a hardware description language (HDL) for an FPGA. Limited power consumption, monetary cost, and practicality will usually drive an SDR design toward a less powerful baseband interface, so the speed increase from a low-level language (an order of magnitude in some cases) will generally be needed to provide the intensive digital signal processing required by the SDR architecture.

Finally, it will also be the job of the baseband processor to interface with a sink for the communications data it is responsible for processing. In the case of this project, an SDR repeater, the baseband processor will be responsible for transmitting the information it has received, possibly using a different protocol stack etc… as well as communicate over a serial interface to a personal computer running a graphical user interface (GUI) front-end for a user to configure the system. Current plans include using Python for GUI development.

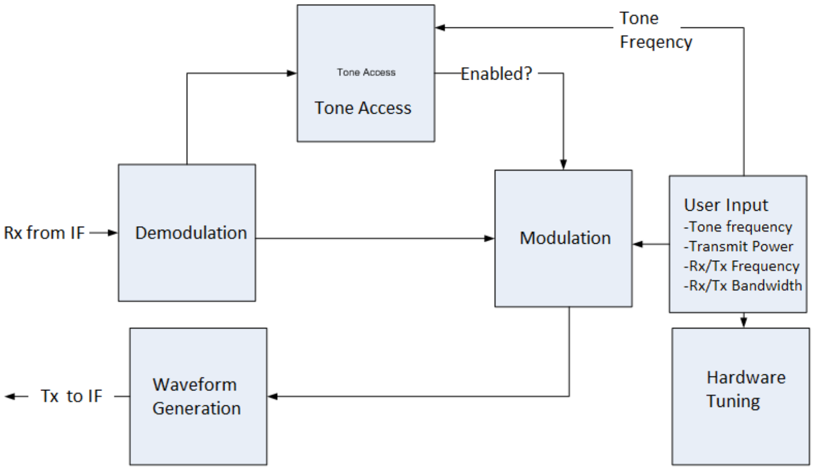

Figure 4: Functional Software Design

Baseband Functionality

In the baseband processing stage of the repeater (shown below), the first module is the demodulation of the signal. The demodulated signal is then fed to both the tone access module and the modulation module. The tone access module looks at the demodulated signal for a constant tone at a certain sub-audible frequency. If this tone is present, the tone access module enables the modulation module. If the modulation module is enabled, it modulates the signal. The bandwidth occupied by this new signal is controlled by the user interface, which also controls the tone access frequency, transmit and receive frequencies, as well as the transmit power.

SECTION 2.2: TECHNICAL DESCRIPTION OF THE DESIGN AND ITS COMPONENTS

With the functionality of each subsystem well-defined, the research shifted toward implementation issues and component specification, as the next step in the development of a hardware design. Section 2.3 will seek to explain some of the mathematics involved in the different performance metrics.

RF/IF FRONT-END

The following components were identified as necessary for implementation for the RF and IF Subsystems. Their basic functions and performance metrics are discussed below.

Analog/Digital Conversion: Because it will impose the most stringent design requirements, the design specification will begin with the ADC. In fact, the ideal ADC for an SDR application does not yet exist. The relatively high sensitivity, bandwidth, and speed requirements of the perfect ADC are not yet realizable practically. For this reason, the choice of ADC will necessitate other design trade-offs elsewhere in the design, such as frequency mixing, filtering, and digital up/down conversion.

Practical requirements for an ADC in this system include: sufficient sampling accuracy to minimize quantization error in a signal, and speed – noting the Nyquist sampling theorem, which states that the maximum frequency signal that can be completely reconstructed will be half the sampling rate of the ADC itself. Note: the as it is of paramount importance, the ADC will be the most expensive single component in the system.

Important measures of performance for ADCs and DACs include:

ADC

- Sample Rate
  (samples/second 100-200 MSPS for SDR)
- Precision (bits, 10-16 for SDR)
- Power Consumption
- Signal-to-noise ratio (SNR)

DAC

- Settling time (ns range for RF)
- Precision (bits)
- Noise figure

RF Mixers: It will fall to the mixers and the PLL frequency synthesizer (or LO) to tune the received frequency (which could be in the range up to 1.3 GHz in some amateur radio bands) to the desired IF frequency, and then to digital up/down conversion to sync the signal with the microcontroller/FPGA used in the digital section.

Relaxing the ADC requirements with the use of an intermediate frequency will defer some of the design decisions to the RF mixers. A good candidate for an RF mixer will need to have bandwidth enough to accept the wide range of RF frequencies in the amateur radio bands, sufficiently low noise figure so as not to degrade the entire system performance, good isolation, especially at the PLL input which may be running at a high power level. Important measures of performance for RF mixers include:

RF Mixer Performance Metrics Include:

- Linearity – IP3 (dB, higher is better)
- Bandwidth (must support the all bands)
- Port Isolation (isolate LO/RF from IF)

Low Noise Amplifier (LNA): Very important to the overall receiver performance is the low noise amplifier, which is responsible for amplifying the relatively low strength signal received at the antenna to increase its signal strength enough to bring it within the sensitivity range of the other RF components, thereby allowing the signal to be processed by them. The low noise amplifier has a huge effect on the overall receiver performance, as its noise figure is the most significant portion of the noise figure of the entire receiver. The implementation schematic shown on the following pages also includes a *gain stage* – which does not have excellent noise performance but does have high enough gain to reasonably amplify an incoming signal for further processing.

LNA Performance Metrics Include:

- Noise Figure (dB as low as possible)
- Gain
- Bandwidth
- IP3 – Linearity

Power Amplifier (PA): Just as the LNA is needed to amplify the low-strength signal at the antenna(e) for system use, the PA will need to amplify the signal coming out of the system so that it can be transmitted powerful enough to give the device range, and allow receivers to discern its output from random noise.

PA Performance Metrics Include:

- High efficiency
- Sufficient gain
- Sufficient bandwidth
- Linearity
- Low out-of-band emissions

Variable Frequency Oscillator (VFO): In order for an SDR radio to tune between different operating frequencies, channels within bands, and be able to mix high RF frequencies to an intermediate frequency, the RF mixers used will need an accurate reference frequency – as they are three port devices (RF, LO, and IF). While tunable local oscillators exist, they usually need to be run at very high power, and have limited range, as well as high noise figure, and in turn produce a number of design trade-offs. A programmable frequency control system known as a PLL can be used to mitigate these problems, as well as allow for completely software-based tuning. The integrated circuit implantation of the PLL is known as a VFO – a variable frequency oscillator.

VFO Performance Metrics Include:

- Low noise figure
- Small channel spacing
- Low Phase Noise
- Wide bandwidth
- Low spurious/out-of-band emissions

RF filters: RF filters, are used for a wide variety of applications in an SDR frontend design. They are generally used to implement frequency division duplexing (FDD), antenna selectivity, and image rejection at mixers. Depending on the application, performance metrics for RF filters can vary widely. They are generally selected for their balance of frequency-domain characteristics with time-domain performance.

RF Filter Performance Metrics Include:

- Cutoff frequency – could be multiple
- Sharpness, or gradual role-off
- Ripple
- Insertion loss (dB)

Matching Networks: Although not very complex, matching networks will be needed to ensure efficient power transfer between components in the system. Usually simple and cheap LC circuits will suffice, values for which can be determined with the use of a Smith chart. They basically function by matching impedance between their two nodes, usually to 50 Ohms in RF systems.

RF/IF Frontend Schematic: The following is the result of much SDR and component research, and consideration of engineering trade-offs. It was developed by group member Matt Schurmann, as part of another course at Stevens, in RF physical layer design for communication systems. Before being built, however, extensive consulting will be done with an advisor to avoid waste of money and time due to unforeseen issues.

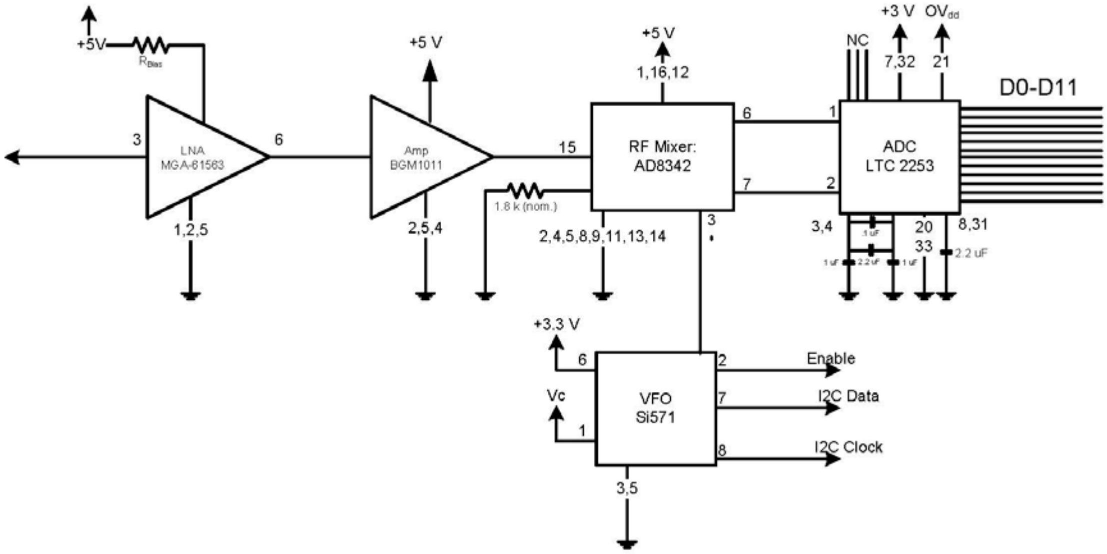

FIGURE 5: RECEIVER SCHEMATIC

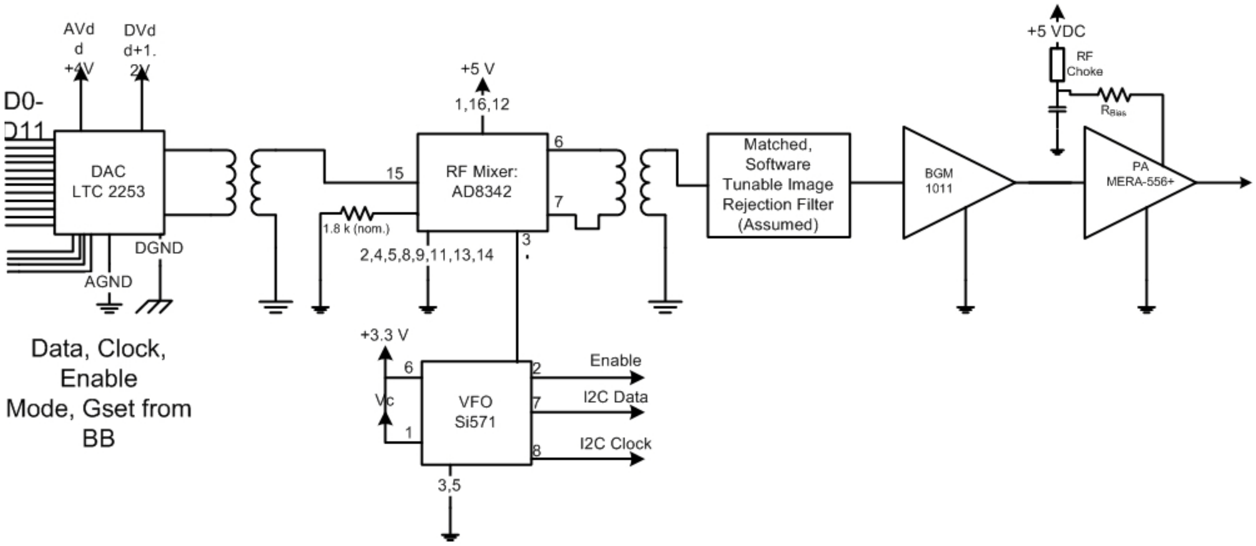

FIGURE 6: TRANSMITTER SCHEMATIC

Part Numbers: The table below summarizes the performance metrics, and models chosen for each component of the RF subsystem schematics shown on the previous page.

Part	Model Chosen	Performance Metrics
ADC	Linear Technology LTC22253	12-bit resolution, 125 MSPS, 70.2 dB SNR, $55.00
DAC	Texas Instruments DAC5662A	12-bit resolution, up to 275MSPS, $30, multiple DAC to allow for software tuning other components
LNA	Avago MGA61513	High bandwidth, 14 dB Gain, 1.6 dB NF, $2.31
PA	Minicircuits MERA 556+	High bandwidth, 20.5 dB, High degree of flatness
Gain Stage	NXP BGM 1011	High Gain (30+dB), reasonable noise figure. $10
RF Mixer	Analog Devices AD8343	Active Mixer, 16.5 dBm IP3, 0-2.5 GHz BW, $5.24
VFO	Silicon Labs Si570	High bandwidth, digital control (I^2C) , phase noise, $15.23

A Few notes about the design that are not immediately apparent from the schematics:

- The RF frontend design will need serious consideration at the time of implementation and may need to be seriously re-worked based on further research
- A viable antenna system and duplexing method will need to be devised at a later date – as the use of time-slotting with a T/R switch, and single antenna may not work well.
- Matching networks are tuned with varactor diodes, which will be connected to cheap, low-resolution ADCs from baseband to tune capacitance digitally
- This design is relatively expensive
- Unlike most RF frontend designs, antenna filtering is handled through digital signal processing, in baseband
- It assumes that a suitable antenna can be found – it does not specify one as of yet

Baseband Processing

FPGA: A Field-Programmable Gate Array (FPGA) is an integrated circuit which consists of programmable logic components and reconfigurable interconnections that allow blocks to be connected. The logic components (logic blocks) can be reconfigured to perform logic gate functions (AND, NOR, XOR, etc.) or more complex functions. Some FPGAs have features like memory elements or analog features.

Applications of FPGAs include DSP, DSR, prototyping, speech recognition, emulation, etc. They are useful in areas that require high performance computing and/or parallelism. FPGAs are effective at computations such as FFT or Convolution. In contrast to CPLDs, FPGAs allow for

FIGURE 7: ALTERA CYCLONE FPGA DEMO BOARD

more flexibility. The disadvantage to using FPGAs is the complexity that is required to design for a specific purpose.

In order to define the behavior of the FPGA the user uses hardware descriptive language (HDL) or a schematic design. The most common HDLs are VHDL and Verilog. The source files are sent to the software suite for the FPGA and then sent to the FPGA via serial interface or from some type of memory medium

The FPGA architecture consists of a matrix of programmable logic elements which are connected through local or global connections. These elements typically operate on 1-bit or –bit words that can be combined together. These elements can be programmed by changing the bit values which are stored in memory elements. This determines the operation of components of the logic element.

GPP: A preprocessor is a system which processes input data and produces an output that is used as an input to something else. A general purpose preprocessor (GPP) is intended to not be limited by a specific task but to be widely functional.

DSP: A digital signal processor (DSP) is a microprocessor that is specialize and optimized for digital signal processing. Typically they are integrated circuits. Most used fixed-point arithmetic and the reduced hardware complexity provides a cost and speed benefit associated with DSPs. DSPs use direct memory access (DMA).

Associated Trade-Offs

There are a multitude of different baseband tradeoffs that need to be made in the implementation of baseband hardware for a software defined radio.

General Purpose Processors:

Pros	*Cons*
• The latency problem in GPPs can be alleviated by employing multi-core processors and by developing a hybrid architecture consisting of an embedded GPP and a reconfigurable FPGA, plus some auxiliary ASICs. • Because of the size of the GPP market and its role in our daily lives, it is fairly commonly known that to avoid significant heat dissipation issues • Peak GPP speeds are significantly greater than DSPs • Current GPPs are fast enough to do a lot of real time digital signal processing tasks and functions. With many library functions and a very friendly development environment, GPPs appear in several widely used SDR architectures, like GNU Radio, OSSIE, Software Communication Architecture (SCA), and Space Telecommunications Radio System (STRS), and can achieve very good reconfigurability. • GPPs have the advantage of an easier development environment and faster reconfiguration than any other platforms, therefore offering the potential of real-time multi-band multi-mode reconfiguration in SDR.	• Latency problems are unavoidable because GPPs are based on the Von Neumann architecture which has a memory hierarchy and OS which introduces run time uncertainty. • The sequential signal processing inherent in GPPs introduces additional execution latency. On the other hand, ASICs, FPGAs, and analog components all process signal in a parallel/pipeline fashion (A continuous sequence of signals is executed simultaneously by a sequential set of components.) • GPPs typically have a much higher consumption than DSPs and FPGAs. • Fundamentally, GPPs can only run one task at a time, even though some instruction and data level parallelisms like instruction pipeline and super-scalar instruction execution are widely implemented.

Field Programmable Gate Arrays

Pros	*Cons*
• FPGAs achieve much greater computational capacities than DSPs • FPGAs are power efficient and to some extent reconfigurable. • ASICs, FPGAs, and analog components all process signals in a parallel/pipeline fashion (A continuous sequence of signals is executed simultaneously by a sequential set of components). • In some high-performance signal processing applications, for example, FPGAs can take advantage of their highly parallel architectures and offer much higher throughput than DSPs. As a result, FPGAs' overall energy consumption may be significantly lower than that of DSP processors, in spite of the fact that their chip-level power consumption is often higher • For ASICs, FPGAs, and DSPs, latency is primarily related to computing capacity which is easy to quantify for radio function’s computational requirement. • FPGAs all execute signals in parallel/pipeline (a continuous sequence of signals is executed simultaneously by a sequential set of components). FPGAs have the advantage of dividing difference slices into different functions, therefore, executing signal processing functions in a pipeline mode.	• FPGAs typically have greater power consumption (due to high static power consumption levels) than DSPs • FPGAs are generally considered the most difficult platform to program and have other practical issues related to the dynamic management of bit images (the means by which FPGAs are programmed). • It is usually very challenging for a single FPGA to support wideband waveforms, even for baseband signals.

<u>DSPs (Digital Signal Processors)</u>

Pros	*Cons*
• For DSPs, latency is primarily related to computing capacity which is easy to quantify for radio function’s computational requirement • DSPs typically use much less power than GPP or FPGAs. • To keep up with more general increases in waveform complexity, DSPs have continued to increase their available computational capacity by increasing the number of simultaneous operations per cycle while increasing clock rates at a slower rate. • While the growth in GPP clock speeds have leveled-off, this has not yet happened for DSPs. • DSPs have also been exhibiting a growing efficiency in terms of the operation rate that can be supported per mW by increasing the number of operations executed per cycle. • Communications DSP chips for signal processing have good flexibility although high performance DSP chips are expensive.	• Latency problems are unavoidable because Digital Signal Processors (DSPs) are based on the Von Neumann architecture which has a memory hierarchy and an operating system (OS) which introduces run time uncertainty. • It is usually very challenging for a single DSPs to support wideband waveforms, even for baseband signals. • DSP processors have their own inefficiencies. In a DSP, only a tiny fraction of the silicon is devoted to computation; most of the silicon area and most of the energy is devoted to moving instructions and data around. Hence, it would be a mistake to assume that FPGAs are inherently less energy efficient than DSPs. • DSPs cannot execute signals in parallel/pipeline

Baseband Design Alternatives

FPGA and GPP: The basic implementation diagram for an FPGA-centric design is as shown below.

The SDR architecture must have similar attributes in reconfiguring as GPPs have, and similar execution speeds as highly optimized ASICs have. SDR should better utilize available computing resources, for instance, increasing more parallelism which is more important than using increased computing ability found in DSP and GPPs. Therefore, the recommended SDR architecture is hybrid.

The hybrid architecture will contain an embedded GPP and a reconfigurable FPGA. Reconfiguration requests by the GPP will prompt the FPGA to reconfigure quickly by using existing function bit streams similar to the way that library functions are used for GPP programs. The architecture is able to reduce the power dynamic by using low clock frequencies which reduces static and dynamic power consumption. There is parallelism at the data, instruction, and task level by spreading out functions between different computing components.

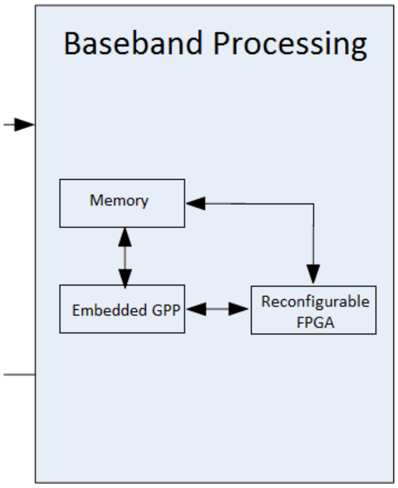

FIGURE 8: FPGA/GPP BASEBAND

DSP and GPP: The basic implementation diagram for a DSP-centric design is as shown below.SDR architecture must achieve a similar ability in reconfiguring as GPPs have, and highly optimized execution speeds, an important aspect in computing ability, which can be found in DSP architecture. Therefore, a potential recommended SDR architecture is a hybrid architecture of both a GPP and DSP.

The hybrid architecture with a control processor may contain an embedded GPP, and a reconfigurable DSP. The GPP is necessary to handle non-DSP functions such as branch control and decisions, and is efficient enough to coordinate different computing tasks. An accelerator is not needed for functions such as graphics processing, as an Ethernet/USB connection will be required for the GPP to communicate with a connected PC, which will handle all visual processing and other tasks. Another requirement is that these processors possess viable DUC/DDC.

This architecture is able to reduce the power budget by using the very efficient digital signal processor, in which technologies have been able to increase the number of simultaneous operations per cycle while keeping clock rates slow. Low clock frequencies save die area and reduce static and dynamic power consumption. One downside of using a DSP rather than FPGA is that a FPGA can achieve parallelism at the data and instruction level and also at the task level by spreading tasks out among different computing components and different FPGA slices, while a DSP cannot.

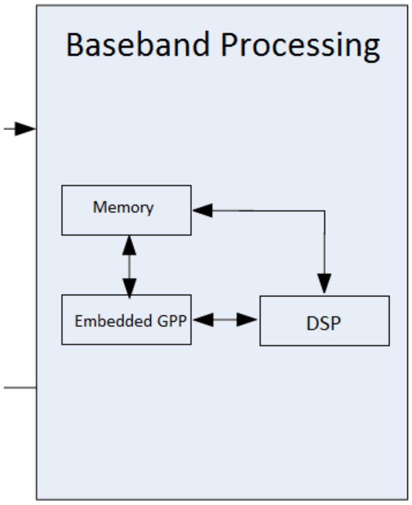

FIGURE 9: GPP/DSP BASEBAND

Most Viable Alternatives

Most likely, the final implementation of the SDR repeater will draw from a slightly modified version of the RF frontend design presented in this report, and a hybrid GPP-FPGA baseband section making use of an FPGA development board with USB and/or Ethernet connectivity. Software will make use of the open-source GNU radio framework.

This choice of architecture should provide an even balance between performance, cost, power, and design time, as well as address a number of the design constraints discussed on the following pages.

Possible Alternative: SDR Kit + GNU Radio

Although this report is primarily concerned with a "from scratch" implementation of an SDR, another avenue for completion of the project has been explored: available for purchase on the market is a product known as the USRP: the "universal software radio peripheral." Similar SDR kits are even available for academic use through SDR research laboratories at Stevens.

Although costly to purchase, and possibly inconvenient to use school equipment through a research lab on campus, this alternative would provide an almost-readymade hardware solution, and allow the group to focus efforts on refining software development. They are usually designed to work with GNU Radio, an open-source software radio framework, which would allow for rapid software development, and allow for more functionality to be added to the radio in the timeframe given design and development. In addition, this alternative would positively impact budgetary concerns, as well as time requirements for the project.

Section 2.3 Mathematical & Other Principles

Analog Modulation

The three basic analog modulation techniques are AM (amplitude modulation), FM (frequency modulation), and PM (phase modulation). The techniques all use a carrier signal. Each technique is named after which characteristic each modifies. For instance, amplitude modulation modifies the property of amplitude of the carrier signal, and frequency modulation modifies the property of frequency of the carrier signal. Each technique has its own advantage and disadvantage.

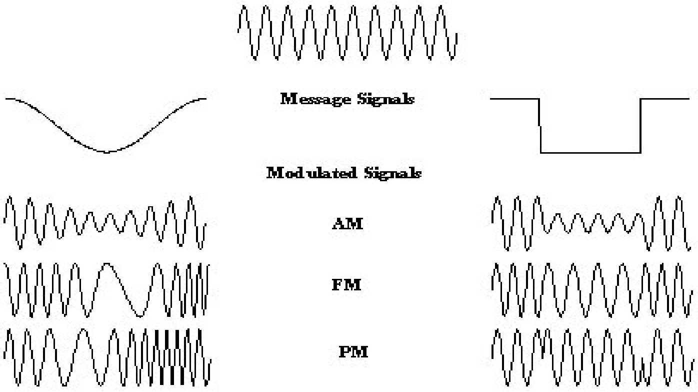

FIGURE 10: ANALOG MODULATION

Amplitude modulation is simple to design but has a couple disadvantages. AM is prone to interference due to noise spikes from the transmission medium. If there is a loss of connection then the digital data that is received are zeros. Frequency modulation has immunity to noise on the transmission medium and there is always a signal present so any loss of signal can be noticed easily. The disadvantage to FM is that it requires 2 frequencies, which makes it more complex, and the detection device needs to be able to recognize both frequencies when the signal is lost. Phase modulation only requires the use of one frequency and it is easy to detect the loss of carrier. The primary disadvantage is that it is complex to generate and detect phase changes.

Mathematical Basis for Frequency Modulation

Frequency modulation can be modeled by taking a signal carrier $v(t) = A\cos\theta(t)$, and modulating the phase $\theta(t)$ such that $\theta(t) = \omega_{IF}t + k\int_{\tau=0}^{t} m(\tau)d\tau$. The frequency is then the derivative of the phase, that is $\omega = \frac{d\theta(t)}{dt} = \omega_{IF} + km(t)$. Modeling the modulating signal as a sinusoid $m\cos\omega_m t$, the phase becomes $\theta(t) = \omega_{IF}t + \frac{km}{\omega_m}\sin\omega_m t$, and the modulated signal becomes $A\cos\left(\omega_{IF}t + \frac{\Delta\omega}{\omega_m}\sin\omega_m t\right)$, where $\Delta\omega$ is the maximum frequency deviation. Fourier analysis shows that the Fourier coefficients are Bessel functions of the first kind. Specifically, the nth coefficient is $J_n\left(\frac{\Delta\omega}{\omega_m}\right)$. Additionally, it can be shown that the bandwidth of an FM signal is approximately $2(\Delta\omega + \omega_m)$. This is known as Carson's rule, and in an amateur system, with a deviation of 5 kHz and a voice signal 3 kHz wide, the bandwidth of an FM signal is 16 kHz. Channels of 20 kHz are typically allocated.

Sampling Theory

We can model the sampling of a signal as the multiplication of the signal with an impulse train, where the rate of the impulses is equal to the sampling rate. In the frequency domain, this corresponds to convolving the signal with an impulse train. This creates a number of identical images that are centered around frequencies that are the sampling rate apart. Thus, it turns out that in order to recover the signal using a low pass signal, the sampling rate must be at least twice the largest desired frequency in the signal.

As mentioned above, this poses a problem for the acquisition of high frequency signals, as acquisition rates are limited by the ADC used. There is another option, however. This is to use under-sampling. If you choose a sampling rate that is at least twice the bandwidth you desire, you create multiple images of the signal at lower frequencies such that there is no overlap. A band-pass filter could easily be used to recover the signal.

DSP: Finite Impulse Response Filtering

As much of the filtering functions required by the RF frontend are to be accomplished in baseband, it is important to have a concept of exactly how a digital signal processor will implement a filter. The most popular method involves the passing of a digitized signal through a difference equation, with a pre-determined transfer function. The transfer function will have the frequency response desired, as the coefficients, order, and degree of the difference equation will be derived mathematically. A logical implementation is shown to the right. It uses a tapped delay line (easily accomplished on a digital signal processor) to delay the signal, and multiply it by a coefficient – the output is tapped for a direct implementation of the difference equation – and thereby a direct convolution with the desired transfer function, providing the desired frequency response.

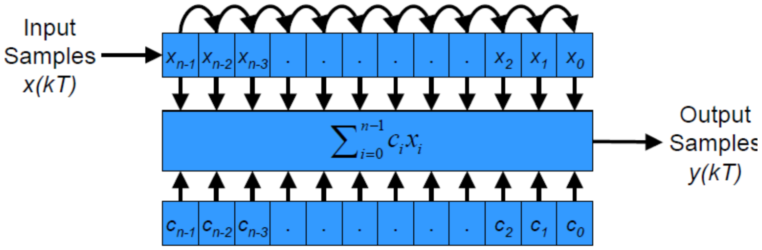

FIGURE 11: TAPPED DELAY LINE FILTER

DSP: The Discrete Fourier Transform

Another task that needs to be performed in the baseband is the analysis of the waveform for the tone access system. This can be accomplished, for example, by analyzing the spectrum of the demodulated signal. In order to accomplish this, the Fast Fourier Transform can be performed on the signal. The Fast Fourier Transform (FFT) is a version of the Discrete Fourier Transform (DFT), which finds the spectrum for a finite length sequence. The N-point DFT is defined as $X[k]=\sum_{n=0}^{N-1}x[n]\exp\left(-\frac{2\pi j}{N}kn\right)$, but this definition is very computationally expensive. For this reason, the FFT was developed. The FFT is actually a collection of optimized algorithms for calculating the DFT, having a computational complexity of O(N log N), as compared to a complexity of O(N^2) for the regular DFT. This optimization is crucial for modern DSP applications. Additionally, since the inverse DFT simply uses the opposite sign and a normalization factor, the FFT is easily adapted to calculate the inverse.

DSP: Direct Waveform Synthesis

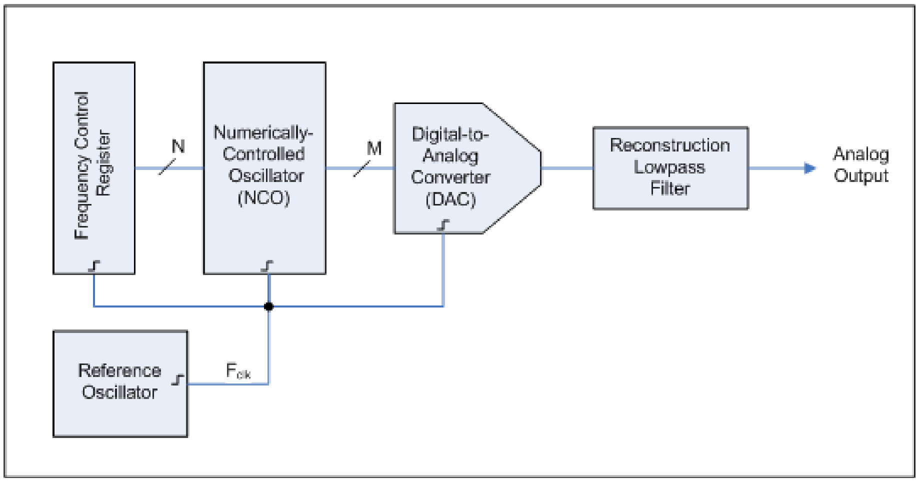
Frequency Control Register
N
Numerically-Controlled Oscillator (NCO)
M
Digital-to-Analog Converter (DAC)
Reconstruction Lowpass Filter
Analog Output
Reference Oscillator
Fclk

FIGURE 12: DIRECT WAVEFORM SYNTHESIS

One of the tasks that the baseband processing needs to complete is the modulation of the signal. For this to occur, it needs to be able to synthesize a carrier waveform in the digital domain. One way to do this is a technique known as direct digital synthesis (DDS). This uses a system clock as a reference to a numerically controlled oscillator (NCO), whose frequency is controlled by a frequency control register. We can therefore modulate a signal by varying the frequency of the control register. For more detail on how to perform FM modulation using this technique.

Section 2.4 Performance Expectations

WE expect to develop a prototype that can successfully operate as a legal amateur radio repeater. This entails meeting certain performance characteristics, which are detailed below.

Noise Considerations

The sensitivity of a receiver is dictated by the amount of noise it introduces to the signal, as well as noise in the surrounding environment. Since we cannot control the environment the system operates in, we want to minimize the amount of noise introduced into the system by the receiver. This is quantified in the receiver's noise factor, which is the ratio of the output SNR to the input SNR. The receiver's noise figure is the noise factor expressed in decibels. A typical VHF receiver has a noise figure of 3 to 4 dB, so that is what we will aim for.

Spurious Responses

Because of nonlinear effects in amplifiers, mixers, and other RF devices, unwanted, or spurious, responses are generated. These spurious signals are often outside of the signals normal bandwidth, and can therefore interfere with other communication channels. It is for this reason that the FCC has mandated that all out of band spurious responses are at least 60 dB below the carrier level. As mentioned above, nonlinearities are the cause of spurious responses. The linearity of a device is quantified by measuring the relationship between the third order harmonic generated by the nonlinearities and the desired signal. This parameter is known as the third order intercept, and is a good indicator of the linearity of a system. By having high third order intercepts and good filters, we can easily achieve the necessary spurious response.

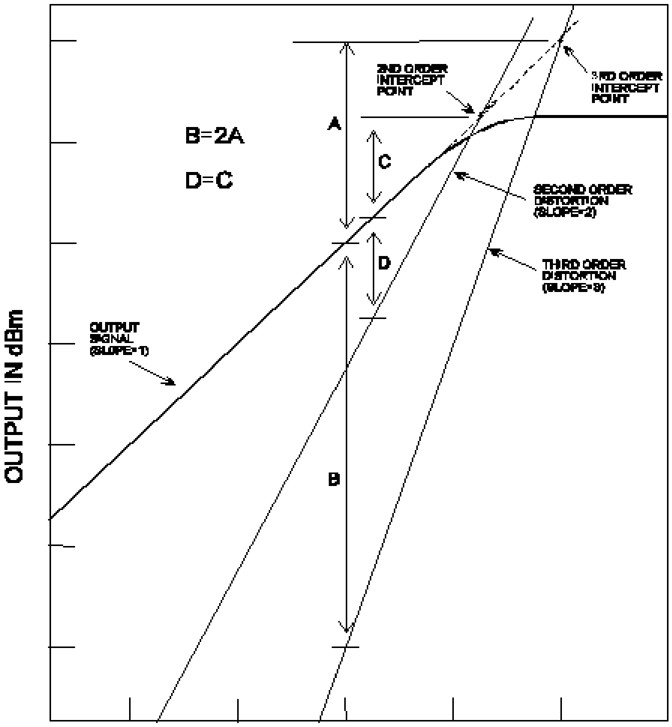

FIGURE 13: ILLUSTRATION OF THE THIRD ORDER INTERCEPT

Selectivity

In order to properly work, the receiver must be able to distinguish the desired signal from neighboring signals. The ability of a receiver to perform this is known as the receiver's selectivity. It is typically measured in decibels, and relates the power levels of rejected signals to the desired signal. A high selectivity is a must for this project.

Image Rejection

When frequency mixing occurs, two signals are converted to the same frequency – one above and one below the local oscillator frequency. The unwanted signal is known as the image signal, and in order to successfully demodulate the desired signal, this image must be rejected by the receiver. This is done using an RF filter before the first mixing stage that passes the desired signal while removing the image frequencies. As an example of what kind of rejection is needed, ICOM's IC-2200H 2-meter transceiver has image rejection capabilities of at least 60 dB.

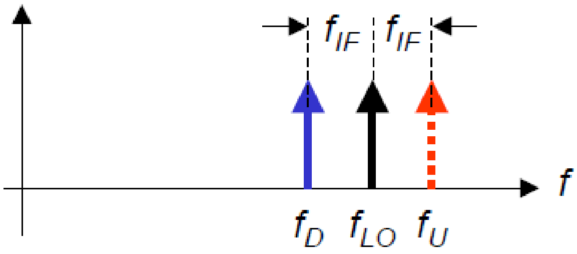

FIGURE 14: IMAGE FREQUENCIES

Transmit Power

In order to reach a large area of operation, the repeater will need a relatively large amount of transmission power. However, since the power needed will depend on where the repeater is in operation, the system will need to be able to handle a large range of transmission powers.

SECTION 3: CRITICAL EVALUATION

SECTION 3.1: THE "GOOD"

The design that is proposed is made from readily available components which can be purchased through consumer websites like digikey.com. Software prototyping will be not be time consuming because GNU Radio software radio framework can be utilized to aid in the area of firmware generation. The currently proposed designs are already feasible and address the engineering constraints associated with the project.

SECTION 3.2: THE "SCARY"

The design is likely to go over budget in one way or other and out of pocket funding may be necessary. Use of open-source software frameworks like GNU may introduce intellectual property issues. Usage of an SDR-kit may be perceived as a very simple and "easy-way-out" solution and creating an SDR design framework will be very time intensive as well as putting the project over-budget. The primary concerns involve the development and funding of the project.

SECTION 3.3: THE "FUN"

This following paragraph speaks not only to our project, but to the very core of what drives the science and engineering disciplines. After all, some may see engineering as a boring profession. How could solving complex mathematical equations be fun? But what these people may fail to understand is that it's not necessarily the individual processes of engineering that make it fun, but the resulting culmination of those processes. It's getting that idea in your head that you want to create something novel and new, that you want to take on a challenge. And then after all the painstaking planning and programming and calculating, when what you imagined finally works the way you envisioned it to work, when that piece of code finally runs or that laser finally shoots through the sky. When you have created that idea in your head, it is what makes it all worth it. This is what makes the process fun, because you know that perhaps you could achieve something great. And every scientist, programmer, and engineer should know this feeling.

SECTION 3.4: FUNDING OF PROJECT

The very small research and development budget given to each group ($250), the process of ensuring that the design is economically viable to manufacture, and the critical requirement that the design be cheaper to maintain/implement than comparable non-SDR repeaters are all economical constraints on the project. Seeing as how it would be impractical for our funding requirements to not exceed $250, some sources of funding may be through applied school and public grants and personal investments. Also, the use of hardware/software development kits and other resources have been offered to us by Stevens Institute, which will help in the next phase of development.

SECTION 4: CONCLUSIONS

In summary, a great deal of research has been conducted on the subject of software defined radio, and the group feels that the implementation of an amateur radio repeater using SDR techniques is quite feasible given the constraints on the project – mainly time, and money. It has been determined that the design will require three basic subsystems, in addition to any support circuitry which has yet to be determined: an RF frontend responsible for receiving, transmission, and frequency mixing, an IF frontend responsible for analog-digital conversion, and a baseband section responsible for tuning RF hardware and performing direct waveform synthesis. The team has yet to decide whether it will be purchasing components, and printing them onto a circuit board to implement the system, or whether it will take advantage of the laboratory resources at Stevens Institute of Technology, to cut costs and relax time requirements.

Although this report is mainly concerned with development of hardware for the system, significant effort will need to be applied to the development of digital signal processing techniques that implement the amateur radio protocol stack, which is, as it turns out relatively simple compared to other communication protocols. One promising avenue for the development of SDR software is GNU Radio, an open-source software radio framework, written in C and Python – languages which members of the team have at least some experience with.

As the project progresses into the implementation and debugging phases over the next year, the research and component specifications described in this report will prove invaluable to the process of design refinement. Although software defined radio design requires a number of complex engineering tradeoffs, and represents an area of uncharted waters for each member of the team, it is a solid, well-established, intensively researched technology, with a plethora of accessible experts and researchers from Stevens Institute of Technology.

This report has described the low-level functionality of the system that will be implemented, as well as a number of the design trade-offs involved. What unforeseen issues will arise as the implementation progresses? How will they be solved? For this, the team will need to continue to employ the problem-solving skills and engineering approach that their excellent education has instilled in them.

SECTION 5: APPENDICES

APPENDIX 1: TEAM MEMBER VITA

Matthew Schurmann is an Electrical Engineering student, Class of 2012, at Stevens Institute of Technology. He comes to Hoboken from Orange County, NY, about 50 miles north of New York City. Matt is an Eagle Scout, and has excellent project and time management skills in addition to technical knowhow in the field of electrical and electronics engineering. A well-rounded engineer, Matt has held technical positions as an RF test engineer intern with Research in Motion, and will be working with Texas Instruments this summer with their Power Engineering group. In addition to engineering positions, he has also been employed as a graphic artist, and a web programmer. His hobbies include web design, everything Linux, listening to music and reading books. See http://mjsch.org for more information.

Scott Curtis is an Electrical Engineering student, Class of 2012, at Stevens Institute of Technology. He was raised in Morris County, NJ about 50 miles from NYC. Scott is a member of several volunteer organizations and has participated in volunteer and leadership positions involving service to the community since he was 14. His primary focuses are in systems engineering, electronics and embedded systems, radio, and computers. He has had an internship experience with TADCO Engineering & Environmental Services, LLC. He has significant hardware lab experience and software experience throughout his rearing. He has earned a Technician class amateur radio license from the FCC. His hobbies include building electronics, playing various instruments, martial arts, and learning.

Erik Thompson is an Electrical Engineering major at Stevens Institute of Technology, class of 2012, where he plans to graduate with a Graduate Certificate in Wireless Communications. He lives in Ramsey, NJ, but spent his formative years in Pleasanton California. Erik is a participant in the institute's co-operative education program, and has held technical positions with Mindray North America, the Automated Test Systems Division of ARDEC, and Safe Flight Instrument Corporation. His hobbies include scuba diving, amateur radio, reading, music, and anime.

Jonathan Pirog is a Computer Engineering Student, Class of 2012, at Stevens Institute of Technology. Raised in Mahwah, New Jersey, he graduated from Mahwah High School in 2007, at which time he also achieved the rank of Eagle Scout. He is a well-rounded "team player" whose core competencies include analytical thinking, efficiency, and creativity. These skills helped him excel in diverse internship positions such as Software Engineering Co-op at Mindray, Cisco systems Analyst at Barclays Capital, and Mobile Innovation Analyst at Merck. This summer he will be working at Barclays Capital in the IT-Platform Support group before returning to school to complete his final year of college. Jonathan's hobbies include competitive gaming, constructing computers, composing music on FL Studio, going to the gym, running, and playing soccer, tennis, and the piano.

Appendix 2: Links to Datasheets

Data sheets used in the RF/IF Frontend schematics presented in the technical portion of this report.

Part Type	**Make/Model**	**Link to Data Sheet**
ADC	Linear Technology LTC22253	http://cds.linear.com/docs/Datasheet/22532fa.pdf
DAC	Texas Instruments DAC5662A	http://focus.ti.com/lit/ds/symlink/dac5662a.pdf
RF Mixer	Analog Devices, AD8343	http://media.digikey.com/pdf/ Data%20Sheets/Analog%20Devices%20PDFs/AD8343.pdf
Gain Stage	NXP BGM1011	http://www.nxp.com/documents/data_sheet/BGM1011_N.pdf
LNA	Avago MGA61513	http://www.avagotech.com/docs/AV02-1471EN
PA	Minicircuits, MERA-556+	http://minicircuits.com/pdfs/MERA-556+.pdf
VFO	Silicon Si570	http://www.silabs.com/Support%20Documents/TechnicalDocs/si570.pdf
FPGA	Altera Cyclone	http://www.altera.com/literature/hb/cyc/cyc_c5v1_01.pdf

Appendix 3: References

Huseyin Arslan and Hasari Celebi, "Software Defined Radio Architectures for CognitiveRadios," in Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems.: Springer, 2007, ch. 4, pp. 109-144

E. Blossom, "Exploring GNU Radio," h*ttp://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html ,* November 2004.

T. J. Kacpura, L. M. Handler, J. C. Briones, and C. S. Hall, "Updates to the NASA Space Telecommunications Radio Systems (STRS) Architecture," in *Software Defined Radio Technical Conference,* Denver, 2007.

J.L Hennessey and D. A. Patterson, *Computer Architecture : a Quantitative Approach*, 3rd ed.: San Francisco, CA : Morgan, Kaufmann Publishers, 2003.

A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*: Wiley; 7 edition, 2004.

Welborn, M.L.; , "Direct waveform synthesis for software radios," *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, vol., no., pp.211-215 vol.1, 1999).

David Pozar. “Microwave and RF Design of Wireless Systems.” Prentice Hall.

Bruce McNair. “EE585WS Class Notes.” Stevens Institute of Technology, Spring 2011.

Sanjit K Mitra. “Digital Signal Processing: A Computer-based Approach.” McGraw Hill. 3e

http://www.arrl.org/files/file/Hambands_color.pdf http://www.eetimes.com/design/automotive-design/4009968/High-speed-ADC-technology-paves-the-way-for-software-defined-radio/

http://focus.ti.com.cn/cn/lit/an/slaa407/slaa407.pdf http://www.xilinx.com/company/gettingstarted/fpgavsasic.htm

http://www.scribd.com/doc/6056669/FPGA

http://en.wikipedia.org/wiki/Field-programmable_gate_array

http://en.wikipedia.org/wiki/Preprocessor

http://en.wikipedia.org/wiki/Digital_signal_processor

http://ossie.wireless.vt.edu/.

http://sca.jpeojtrs.mil/

http://preview.rfdesign.com/mag/601RFDF3.pdf

http://ic.ncue.edu.tw/Seminar_new/9402/950314/950314_02.pdf

http://en.wikipedia.org/wiki/Data_Link_Layer

http://www.icomamerica.com/en/products/amateur/mobile/2200h/specifications.aspx

http://www.matrixtest.com/Literat/mtn109.htm

http://www.k0nr.com/rwitte/2m_frequencies.html

http://www.rigacci.org/docs/biblio/online/intro_to_networking/c4014.htm

http://en.wikipedia.org/wiki/Discrete_Fourier_transform

http://en.wikipedia.org/wiki/FFT

http://en.wikipedia.org/wiki/Direct_digital_synthesizer

http://www.arrl.org/part-97-amateur-radio

http://en.wikipedia.org/wiki/Selectivity_(electronic)

http://mjsch.org/d6/files/EE585.pdf

http://mjsch.org/d6