STEVENS INSTITUTE OF TECHNOLOGY

SDR Amateur Repeater

Hardware Implementation

Erik Thompson, Matt Schurmann, Scott Curtis, Jon Pirog

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

4/8/2011

This document provides a rough hardware implementation design for the development of a Software Defined Amateur Radio Repeater. It was developed during the Spring 2011 semester as a part of the Engineering Design program at Stevens Institute of Technology, and provides detailed information about requirements, specifications and design considerations.

Contents

Team Member Responsibilities 2
RF/IF Front-End 3
Introduction to Components and Functions 3
RF Frontend Implementation 5
Baseband 7
Introduction to Components/Functions 7
Associated Trade-Offs 8
Design Alternative #1 – FPGA and GPP 10
Design Alternative #2 – DSP and GPP 11
Most Viable Alternatives 12
Possible Alternative: SDR Kit + GNU Radio 12
Design Constraints/Future Planning 13
SWOT Analysis 14
References 15
Appendix #1: Data Sheets 16
Appendix #2: Modified Functional Block Diagrams 16

Team Member Responsibilities

Erik Thompson:
- Multidisciplinary Planning

Matt Schurmann:
- RF Frontend Components/Functions
- RF/IF Frontend Design
- SWOT Analysis
- SDR Kit + GNU Radio Alternative

Scott Curtis:
- Baseband Components/Functions
- Baseband Design #1
- Ethical and professional responsibilities

Jon Pirog:
- Baseband Trade-offs
- Baseband Design #2
- Design constraints

RF/IF Front-End

Introduction to Components and Functions

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:

- Linearity – IP3 (dB, higher is better)
- Bandwidth (must support the entire amateur radio band)
- 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.

Important measures of LNA performance 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.

A well-performing power amplifier will need to have a few characteristics:

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

Phase-Locked Loop Frequency Synthesizer (PLL)/Tunable Local Oscillator (LO)

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.

Desirable attributes of PLL/LOs 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.

Performance metrics include:

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

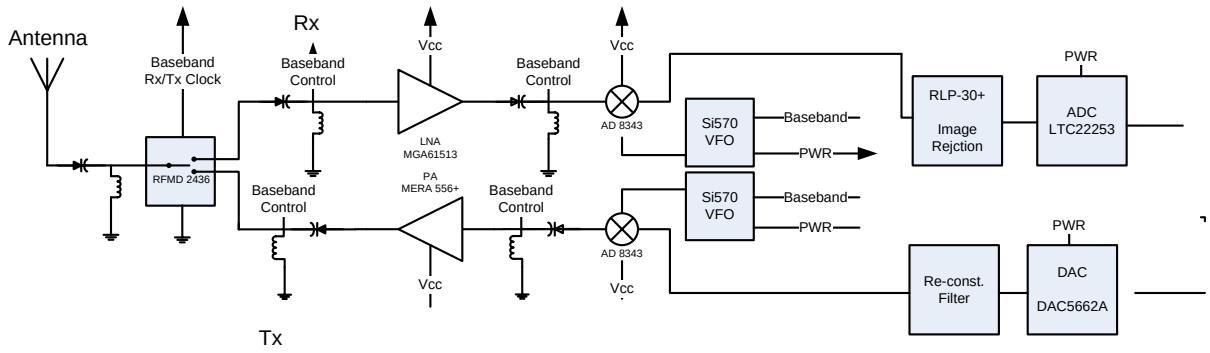
Matching Networks

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 Frontend Implementation

This design follows a somewhat more detailed block diagram than that given in homework 6, see appendix 2.

Schematic



Part Numbers:

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
RF Mixer	Analog Devices AD8343	Active Mixer, 16.5 dBm IP3, 0-2.5 GHz BW, $5.24
Image Filter	Minicircuits RLP-30+	Favorable RF filter characteristics for image rejection
RF Switch	RFMD 2436	High bandwidth, low insertion loss, low VSWR
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 schematic

- 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 design, 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

Introduction to Components/Functions

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.

Altera Cyclone II FPGA Dev Kit

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 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 – "General Purpose Processor"

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 – "Digital Signal Processor"

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

GPPs (General Purpose Processors)

Pros

- 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 that of 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.

Cons

- 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.

FPGAs (Field Programmable Gate Arrays)

Pros

- 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.

Cons

- 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.

DSPs (Digital Signal Processors)

Pros

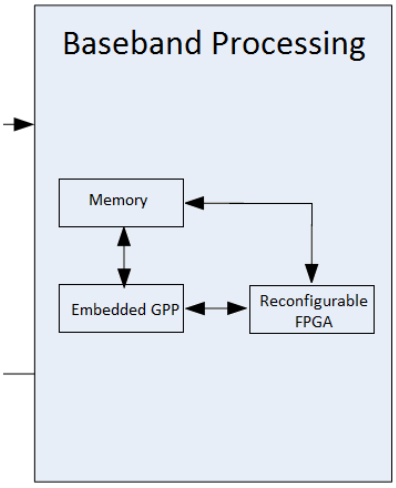
- For DSPs, latency is primarily related to computing capacity which is easy to quantify for radio function's computational requirement
- DSPs typically use much less power than GPP or FPGAs.
- To keep up with more general increases in waveform complexity, DSPs have continued to increase their available 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.

Cons

- 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

Design Alternative #1 – FPGA and GPP

The basic implementation diagram for an FPGA-centric design is as follows:

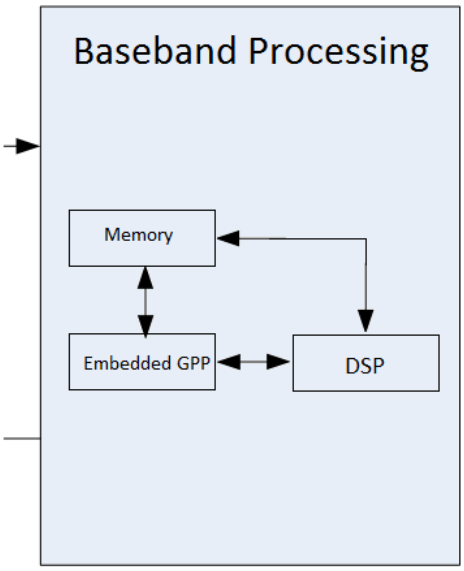


The SDR architecture must have similar attributes in reconfiguring as GPPs have, and similar execution speeds as highly optimized ASICs have. SDR should better utilize available computing resources, for instance, increasing more parallelism which is more important than using increased computing ability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

Design Alternative #2 – DSP and GPP

The basic implementation diagram for a DSP-centric design is as follows:



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.

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.

Design Constraints/Future Planning

Multidisciplinary Planning

There are two main multidisciplinary aspects to the project. The first is the design of the case of the repeater. A mechanical engineer could analyze the case from a heat transfer perspective to determine what kind of heat sinking and cooling functions will be needed. The engineer could also design the mount points for the case internals as well as the external mounting points. The second aspect is the software design, particularly the design of the user interface. For this we could use the help of a computer scientist. The CS could also help with the scheduling aspects of the embedded system.

Technical Constraints

Technical Constraints for this project include**:** compliance with FCC rules and regulations, standards compliance for every protocol stack that the prototype implements, keeping power consumption to a reasonable level for practical use, and bandwidth considerations.

Economic Constraints

Economic constraints for this project include: the very small research and development budget given to each group ($250), 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. 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.

Environmental Constraints

Environmental constraints for this project include: controlling unwanted spectrum access (harmonics, noise suppression), and ensuring that none of the required manufacturing/design processes involved in production are environmentally damaging. This should be a non-issue as all parts are pre fabricated within the guidelines of FCC regulation, and the context of the project does not warrant concern for environmental threat.

Health & Safety Constraints

Health and safety constraints for this project include: keeping design and manufacturing processes safe, design engineers (group members) safe from possible electromagnetic radiation health concerns, and once again, FCC regulation compliance (for legal safety)

Manufacturability Constraints

Manufacturability constraints for this project include: using common and easily reproducible components and manufacturing processes, and keeping the physical layer design as modular and easily maintainable as possible. Resources for the various parts such as needed GPP, ADC, or DSP components, for example, are readily available on the market.

Sustainability Constraints

Sustainability constraints for this project include: the hardware must be powerful enough to handle software algorithms needed for future (yet undeveloped even) protocol stacks, and making the software modular enough so as to make it easily upgradable and maintainable. As for concerns of future sustainability, improvements in data converter performance have been driven by introductions of new architectures and Moore's Law, which implies that ADC performance should continue to steadily improve for the next 16 years. Solutions such as multiple processors and parallelism help drive computational efficiency while keeping power consumption low.

SWOT Analysis

These SWOT points are meant to add to those listed in previous reports, please refer

Strengths –

- Use of the GNU Radio software radio framework will allow for rapid software prototyping and time savings in the area of firmware generation
- The designs proposed are made from readily available components, many of which are purchasable through sites like http://www.digikey.com
- The designs proposed do a good job of feasibly addressing the engineering constraints on the project

Weaknesses –

- Use of open-source software frameworks such as GNU radio may introduce intellectual property issues in the event
- Development of a "from scratch" SDR design will be very time intensive, and surely put the project over-budget, whereas use of an SDR kit runs the risk of being perceived as an "easy way out" solution
- The designs proposed will all run the project over-budget in one way or another

Opportunities –

- Since there will really be no hardware design differences involved, the functionality of the radio as a repeater could be extended to general purpose use, time permitting.

- The introduction of interdisciplinary engineers to the project could lead to synergy that will enhance the project's readiness to solving and predicting issues that could arise.

Threats –

- None of the group members have real-world experience with PCB layout which will certainly be an issue given the dependence of RF circuitry on layout, limited budget, and limited timeframe of the project. If multiple passes of a board are necessary, the project could be easily derailed.
- The tight budget and timeframe of this project will leave it vulnerable to unforeseen circumstances

References

- 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
- 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.
- http://ossie.wireless.vt.edu/.
- http://sca.jpeojtrs.mil/
- 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.

Appendix #1: Data Sheets

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
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
RF Switch	RFMD 2436	http://www.rfmd.com/CS/Documents/2436DS.pdf
FPGA	Altera Cyclone	http://www.altera.com/literature/hb/cyc/cyc_c5v1_01.pdf

Appendix #2: Modified Functional Block Diagrams