Design of a Mobile Transceiver for Precision Indoor Location

by

Matthew C. Campbell

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering
by

November 2010

APPROVED:

Professor R. James Duckworth, Major Advisor

Professor David Cyganski

Professor John Orr

Abstract

This thesis documents the design and implementation process for the next generation of the WPI Precision Personnel Location (PPL) system hardware. The driving goal of the new hardware was to support a new method of radio frequency location developed at WPI referred to as Transactional Array Reconciliation Tomography (TART). This new method is based on a time of arrival (TOA) technique as opposed to the previous Singular Value Array Reconciliation Tomography (σART), which uses time difference of arrival (TDOA). The use of a TOA method requires additional timing information and necessitates a bidirectional (transmit and receive) multicarrier transaction. The design of the new transceiver that can function as both a mobile locator and a static reference unit is the main focus of this thesis. This redesign also addressed previous hardware issues that have been exposed through extensive use in real world testing.

Acknowledgements

I would like to thank my advisor, Professor Jim Duckworth, for all the help and understanding he has given to me during my time with the precision personnel location project. I would also like to thank Professor David Cyganski for his help and input.

I would like to thank all the PPL team members, both current and past. Working with these talented individuals has been a pleasure both intellectually and socially. I would like to specifically thank Vincent Amendolare for his help during my time with the PPL project.

Most of all, I would like to thank my family not only for the lifetime of support and encouragement, but also for the extra level of help during the writing of my thesis. This thesis would not have been possible without my parents, my brother, and my aunt.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis was written in support of the Precision Personnel Location (PPL) project that is being conducted in the Electrical and Computer Engineering department of Worcester Polytechnic Institute. The goal of the PPL project is to produce a field-ready system that will allow precision location and tracking in GPS-denied and indoor environments. Several major milestones have been reached prior to the topics that this thesis deals with, such as a real-time demonstrations of submeter accuracy location and tracking. Future work that is needed to bring the hardware to a field-testable state is documented to provide a road map for future researchers on this project.

This thesis documents the design and implementation of the newest revision of hardware that will support the PPL system. Transactional Array Reconciliation Tomography, or TART, is a new location method recently developed as part of the PPL project. All previous RF approaches of the PPL project have been based on unidirectional transmissions of multicarrier waveforms. TART is based on a bidirectional method that requires both transmit and receive capabilities at all units throughout the system. The need for new hardware to support this method is the driving factor behind this thesis and the work that it documents.

Several acronyms are used throughout this document for brevity. Each is defined within the text at first use; however, a complete list is shown in Table 1.1 for reference.

ADC	Analog-to-Digital Converter
AQM	Analog Quadrature Modulation
CAMP	Control and Messaging Protocol
DAC	Digital-to-Analog Converter
DC	Data Channel
DDR	Double Data Rate
DWG	Digital Waveform Generator
ERP	Effective Radiated Power
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
HDL	Hardware Description Language
IFFT	Inverse Fast Fourier Transform
LO	Local Oscillator
LVDS	Low Voltage Differential Signaling
MCU	Microcontroller Unit
PLL	Phased-Locked Loop
PPL	Precision Personnel Location
PROM	Programmable Read-Only Memory
PSM	Psychological Status Monitoring
RF	Radio Frequency
RX	Receiver
σART	Singular Value Array Reconciliation Tomography
SDR	Single Data Rate
SNR	Signal to Noise Ratio
SRM	Small Radio Module
TART	Transactional Array Reconciliation Tomography
TCVR	Transceiver
TDM	Time Division Multiplexing
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TX	Transmitter
UHA	Unified Hardware Approach
VCO	Voltage Controlled Oscillator
VHDL	VHSIC Hardware Description Language
WPI	Worcester Polytechnic Institute

Table 1.1: Acronyms

1.1 PPL Project History

On December 3rd, 1999, six firefighters tragically lost their lives while battling a fire at the abandoned Worcester Cold Storage and Warehouse in Worcester, Massachusetts[7]. It is possible that many, if not all, of those deaths could have been prevented if a system for tracking and locating first responders had been available.

After hearing of this incident, Professor John Orr, Department Head of Electrical and Computer Engineering at WPI, felt that research into indoor location and tracking could result in just such a system. In December of 2001, Orr authored a white paper outlining the requirements for tracking first responders inside buildings[8]. It is from this white paper that the PPL research project was started.

The goal of the PPL project is to produce a field-ready system that can locate and track first responders in hazardous indoor environments. Figure 1.1 shows a precision personnel location system concept. With input from the first responder community and years of work and research, the PPL project has come to encompass the needs and requirements for such a system.

At the heart of the system is a simple computer interface used by the incident commander. This interface must convey in real time the locations and paths of first responders inside the hazardous indoor environment. Additionally, it must be easy to use and be able to dynamically work with changing situations.

Other requirements are effortless deployment and zero pre-installed infrastructure. The job of first responders is a dangerous and time-sensitive endeavor, and the use of a tracking system must not hinder them by introducing complicated and time-consuming setup. Fires can occur at any location, so there is no guarantee of a pre-installed infrastructure for location and tracking. For this system to preform reliably and in all necessary situations, it must deploy easily and work as a complete self-contained unit.

In order for the system to be useful, the location information must be accurate. Although the difference of just a few meters might seem inconsequential, those few meters could place a wall or an entire floor between a rescue party and the downed firefighter. For that reason, it is the goal of the PPL project to maintain submeter accuracy.

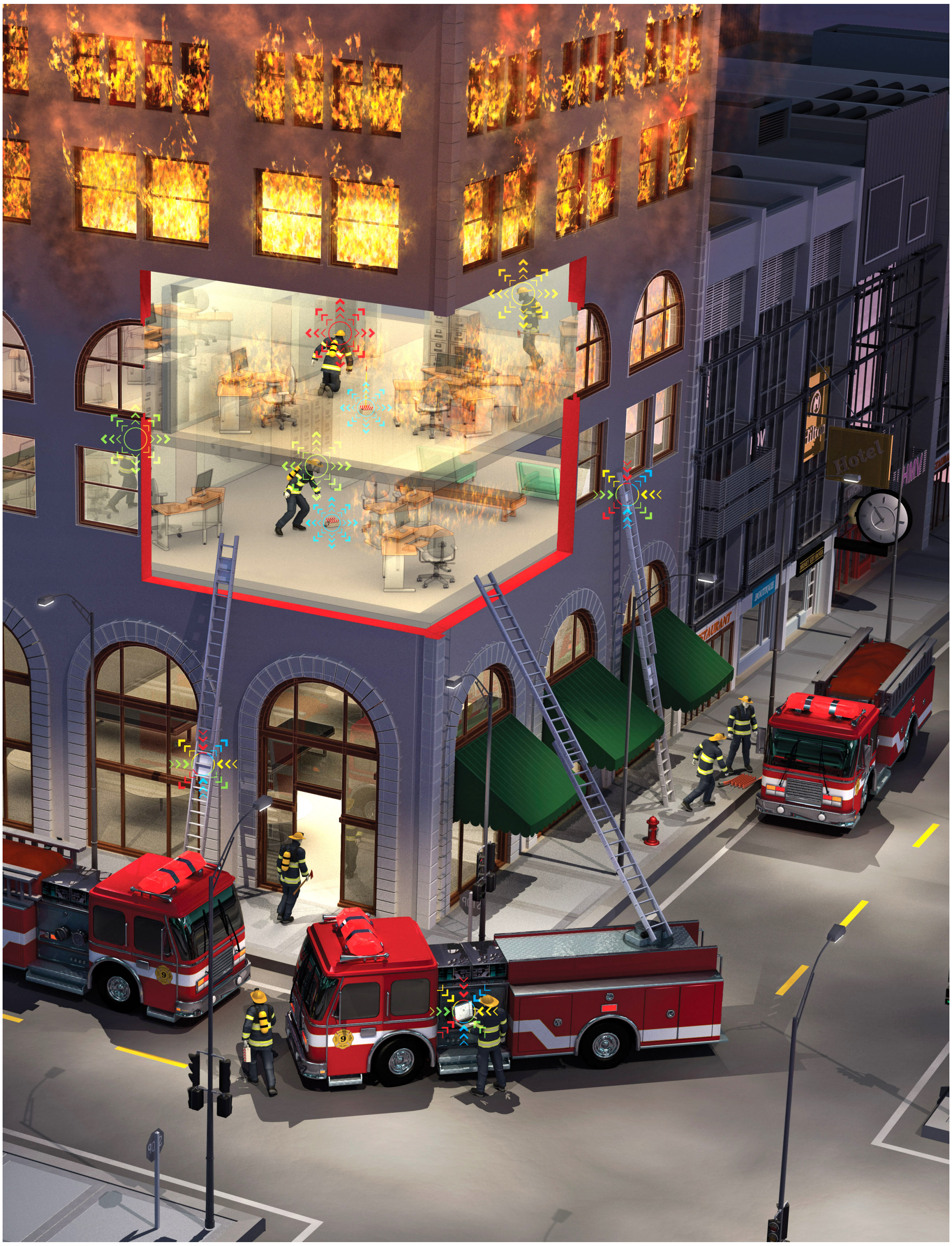


Figure 1.1: Precision Personnel Location System[1]

1.2 PPL Methods

From its beginnings, the PPL system has been primarily an RF-based approach to indoor location. An RF approach was initially chosen for its low cost of implementation and drift-free nature. Though some changes in the system's configuration have occurred through years of research, several core aspects have remained. In addition to these constants, several additions to the system have been made to improve its performance and usefulness to the first responder community.

Central to the RF system is a wideband multicarrier signal. This signal consists of a number of evenly spaced sinusoids. The band of transmission has changed over the years due to different FCC licenses, but the characteristics of this signal have remained mostly unchanged.

The mathematical algorithms that are used for location estimation have changed as research progressed. Location estimation methods include a direct state space approach, Singular Value Array Reconciliation Tomography (σART), and the current state of the art method: Transactional Array Reconciliation Tomography (TART). These algorithms started as proof of concept and have evolved into sophisticated real-time tracking and location systems.

In addition to the use of RF signals for location, the PPL project has explored other sensors for supplementation. Chief among these is the use of high precision inertial sensors. These sensors track the acceleration forces acted upon them and can produce position estimates through processing. However, inertial sensors suffer from open loop drift due to the nature of this method and quickly accumulate error. When paired with the RF system, inertial methods have proven to be a good supplementation for location and tracking. Additionally, barometric sensors have been explored as a possibility for increasing the accuracy of height estimates.

One other noteworthy augmentation to the PPL system is the addition of physiological monitoring systems. These monitors are a collection of personnel-mounted sensors that can measure vital signs, including heart rate, oxygen blood saturation, and respiration rate. Two such sensors are the Foster-Miller Physiological Status Monitoring (PSM) system and the WPI-designed Pulse Oximeter. These systems allow for real-time monitoring of a first responder's vitals in a hazardous environment so that potential health-related incidents can be averted.

1.3 Outline

This thesis begins with a background discussion of all previously used locator hardware. This information provides insight into the evolutionary process that has produced the current revision of locator hardware. The mathematical algorithms used by the PPL project are covered in moderate detail to give the necessary background to understand the hardware implementation. Following this, the design process for the current revision hardware is documented. Requirements, component selection, and system design are all covered. At the heart of this system is a reconfigurable Xilinx field programmable gate array (FPGA). Several more-complicated aspects of the FPGA design are covered in more detail. Additionally, the design and system integration of the transceiver components are explained. At the time of the writing of this thesis, the locator was not yet in a state ready for field testing. The system testing conducted on the hardware as well as a current status of the various subsystems is covered in an additional chapter. Finally, conclusions about the design process are presented along with future work that is needed to bring the hardware to a field-testable state is documented to provide a road map for future researchers on this project.

This thesis builds upon the work of previous PPL team members. In particular, Vincent Amendolare's PhD research relating to TART is a key component in the new system functionality[3]. Additionally, the schematic capture and PCB layout were performed by an external consultant to reduce implementation and construction time for the PPL project. The overall system requirements, component selection, and design and implementation of the software, firmware, and HDL design of the FPGA systems, and peripheral component interfacing, were the major unique contributions to the new mobile transceiver.

Chapter 2

Background

Before delving into the design and implementation of the locator hardware, certain background knowledge is necessary. The PPL project has been an ongoing effort for many years and has seen several revisions of locator hardware. Much of the current design is based on previous hardware revisions and improved from new information acquired through research and testing with these revisions.

Additionally, a theoretical background covering the location algorithms is necessary to understand the hardware implementation. Although the current hardware is designed to support the new TART location method, a background in the previous σART method is also provided to help the reader fully understand the concepts of the PPL project's location methods.

2.1 Time Line of Hardware Revisions

Throughout the life of the PPL project, several revisions of hardware have been created to support the ongoing research effort. Initially, the idea of a multicarrier location system was demonstrated with an acoustic ultrasonic proof of concept. From there, the system has gone though several revisions, which are summarized in the following subsections. Four total RF-based revisions have been produced, starting with revision 0 and culminating with the current revision 3 hardware documented in this thesis.

2.1.1 Revision 0

The initial RF hardware for the PPL project (see Figure 2.1) was created to support a proof of concept. This hardware was a mix of off the shelf and PPL-designed components. In this revision, a multicarrier RF signal was transmitted that differs from the current system used. A total bandwidth of 25–50MHz located from 410–470MHz was used for testing with this hardware.

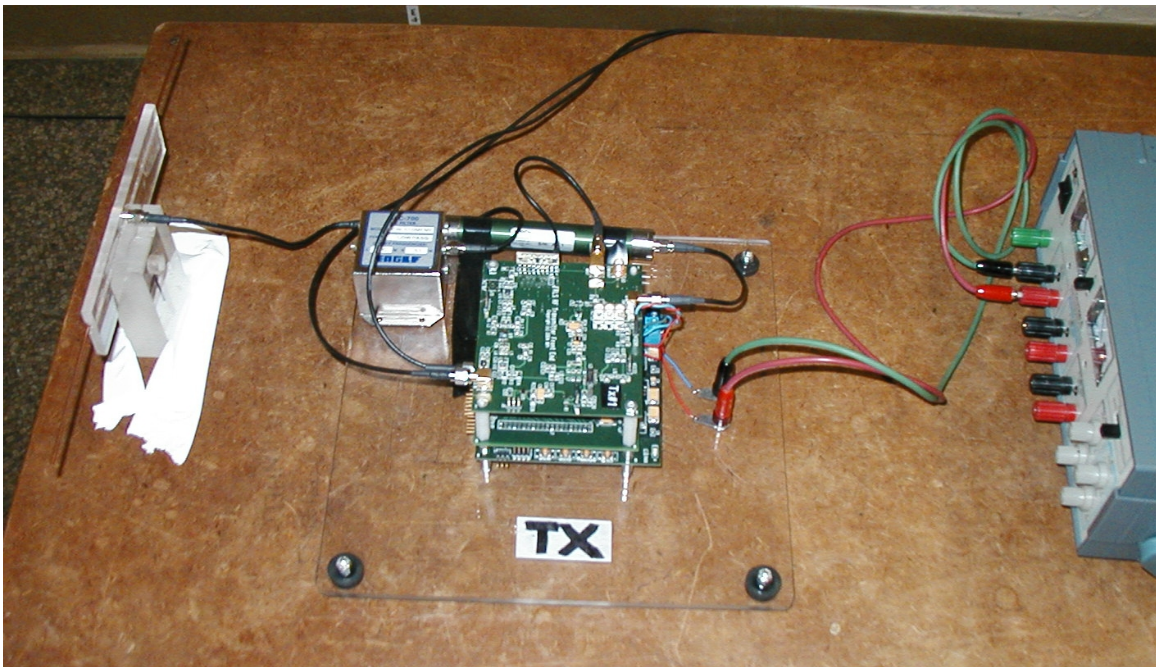


Figure 2.1: Revision 0 Transmitter Hardware

The hardware was mounted on Lexan boards and powered by laboratory bench-top power supplies. These factors made the hardware highly immobile and cumbersome to use for running experiments. Additionally, the architecture of the hardware was highly specific to the testing at hand and, compared with more recent revisions, was far less flexible in supporting changes in research direction. However, this hardware served its purpose of showing the potential of wideband multicarrier RF-based location methods.

2.1.2 Revision 1

Revision 1 represents the first generation of mobile hardware (see Figure 2.2) and was introduced in the second half of 2006. This design was a big step forward because all previous testing had been done with board-mounted proof-of-concept hardware. The locator revision was a compact, battery-powered system that was contained in a custom personnel-mountable case. All the RF systems, signal generation, and overall control were managed by a newly designed MSP430-based embedded system.

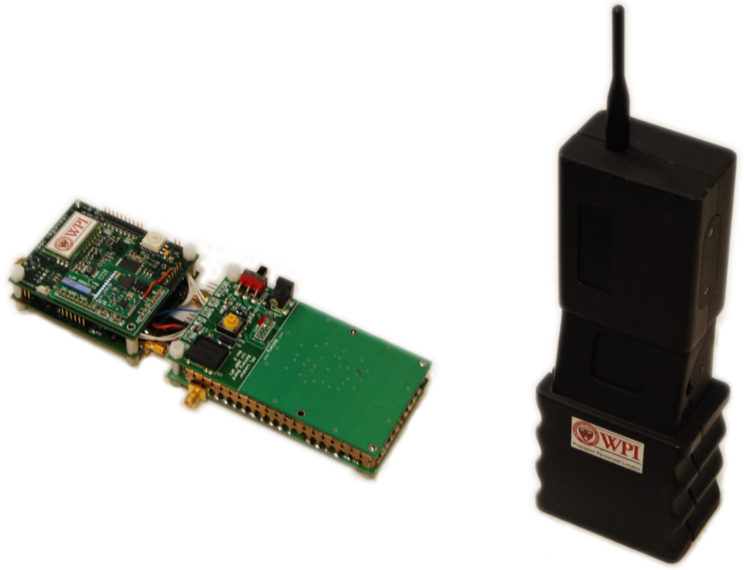


Figure 2.2: First Generation Mobile Locator Hardware

The hardware in this system was divided into three printed circuit boards (PCBs), each with a specific task (see block diagram in Figure 2.3). The three sections were the digital waveform generator (DWG), RF front end, and the data channel.

The DWG was responsible for producing a baseband PPL multicarrier signal. In revision 1 of the hardware, approximately 150MHz of usable bandwidth was required. In addition to simply producing the signal, this unit needed to be able to produce arbitrary signals without hardware modification to adapt agilely to the needs of the PPL project. The core of this board was the pairing of a Spartan3 FPGA with a 400MSPs Texas Instruments digital-to-analog converter (DAC) (over-

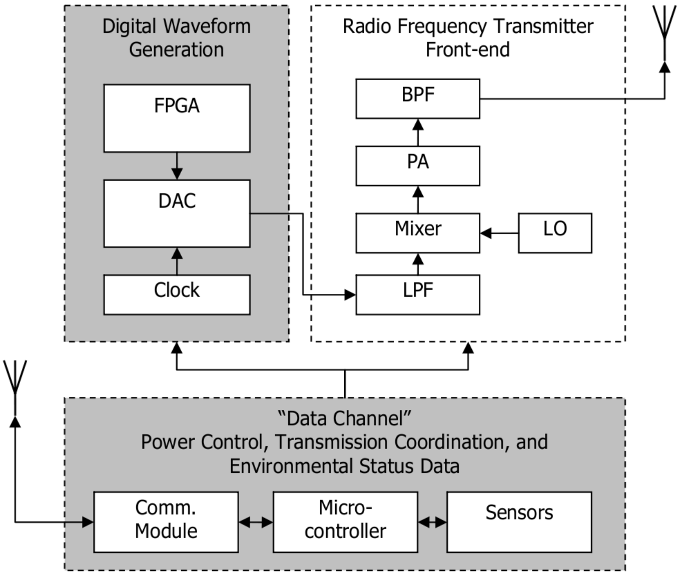

Digital Waveform Generation
FPGA
DAC
Clock
Radio Frequency Transmitter Front-end
BPF
PA
Mixer
LO
LPF
"Data Channel"
Power Control, Transmission Coordination, and Environmental Status Data
Comm. Module
Micro-controller
Sensors

Figure 2.3: Revision 1 Locator Block Diagram[2]

clocked to 440MSPs). The digital samples were stored in the FPGA and fed at high speed to the DAC. Any signal could be transmitted by simply updating the programmable read-only memory (PROM) image of FPGA to contain new samples in RAM.

The RF front end board was responsible for taking the entire baseband (0–220MHz) signal range, which contains the 150MHz of usable bandwidth, and conditioning it for RF transmission. The first step was a low pass filter to remove imaging from the DAC's baseband output. The signal was up-mixed with a 730MHz local oscillator. The final signal was then amplified, and a final band pass filter was applied to suppress any out-of-band noise that had been amplified.

The implementation of this mobile hardware was a great step forward for the PPL project. Its use continued for almost three years before another major hardware revision was implemented. During that time, many theoretical innovations and impressive demonstrations were carried out that otherwise would have been impossible.

In-depth documentation of this design can be found in a past thesis produced by the PPL project[2].

2.1.3 Revision 2

The revision 2 hardware consisted of several changes from the previous revision. These changes are present not only in the mobile locator hardware but also in the reference stations used for receiving the RF ranging signal. This new set of hardware introduced a new overall PPL system architecture.

The system architecture for the revision 2 hardware is shown in Figure 2.4. There are three major components of this system architecture: the mobile locators, the reference transceiver devices, and the base station.

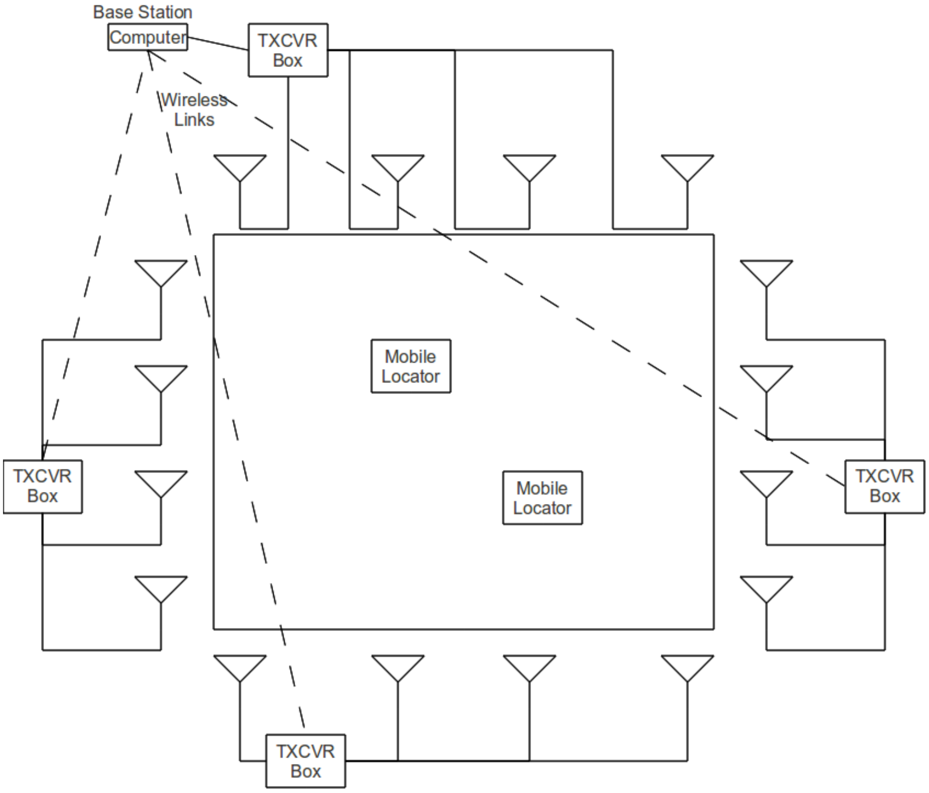


Figure 2.4: Revision 2 System Architecture

Around the perimeter of the scan area, four reference stations, each consisting of one of the revision 2 transceiver boxes, are placed to provide adequate coverage and geometric diversity. Hardwired to each transceiver box is an array of four RF antennas. The transceiver hardware in this setup is used primarily for receiving the signals transmitted by the personnel mounted mobile locators, which transmit in a time-multiplexed fashion. However, its transmission capabilities are used during timing synchronization among the four transceiver boxes. Each of the transceiver boxes is connected with a wireless data link to the base state. The base station is also co-located with one of the four transceiver boxes, which is hardwired rather then connected via wireless link.

The base station, which represents the incident command center of a fire scene, is connected with all other hardware in the system over wired and wireless interfaces. The incoming data is fed into a processing computer where the PPL location methods are carried out and displayed.

The mobile locator and the transceiver boxes are the key components of the revision 2 PPL system. The following sections describe this custom PPL hardware used in the the system architecture in greater detail.

Mobile Locator Hardware

In the revision 2 mobile locator hardware, a departure from the previous generation was manifested in the use of a Spartan3A FPGA at the core of the digital system. The replacement of the MSP430 microprocessor with an FPGA allowed for extreme flexibility and higher computational power. In addition to the use of an FPGA, several other advances were made to this revision, including the addition of various peripheral sensors, an all new RF system design, and a smaller PCB size. All of these aspects contributed greatly to the platform's success as a versatile research vehicle and proving ground for new PPL concepts.

One other change from revision 1 was a new partitioning of the hardware. The DWG and RF front end were modified and combined into a single space-saving PCB (the bottom block in Figure 2.5). While the DWG portion of the design remained mostly unchanged, the RF portion was redesigned from the ground up. This new RF hardware allowed for improved power consumption, higher power output options, and an overall smaller PCB footprint through combining with the DWG design. The final PCB for the DWG/XMTR can be seen in Figure 2.6.

The second PCB in the revision 2 hardware, the top block in Figure 2.5, was the Spartan 3A FPGA-based data channel, which was responsible for system management and digital control. The DWG/XMTR board and the data channel were designed to be the same size and were positioned in a stack with each other (Figure 2.7). These two segregated boards operated independently, sharing only power and four digital IO lines.

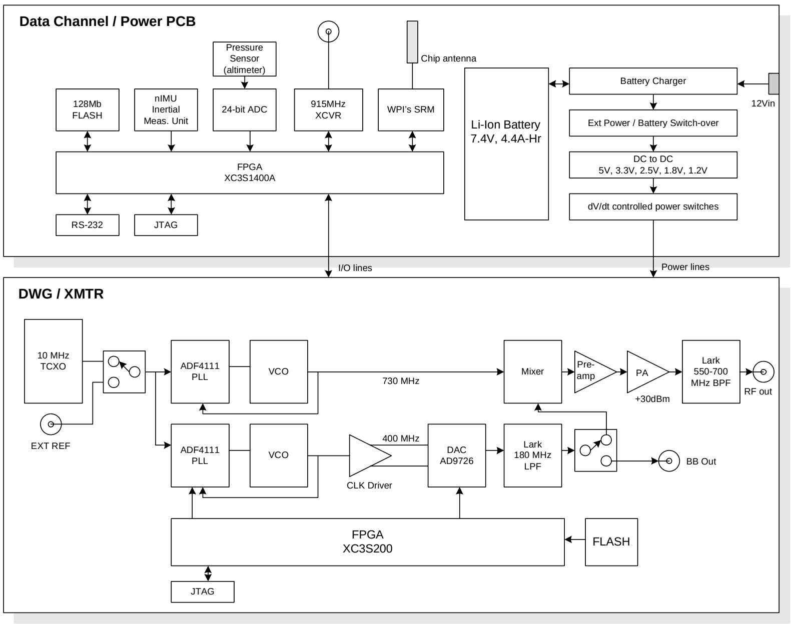


Figure 2.5: Revision 2 Block Diagram

At the core of the FPGA-based data channel is Xilinx's softcore 32-bit Microblaze microcontroller unit (MCU), which is implemented on the FPGA. In previous designs with the MSP430, the lack of flexible hardware I/O proved to be a limiting factor in quickly implementing new peripherals. For example, when the MEMSense nIMU inertial unit was introduced to the project, the lack of an I^{2}C interface necessitated a software implementation of the protocol. This software implementation monopolized all of the MSP430's processing resources, requiring an additional data channel to allow for experimentation with the nIMU device. With the FPGA-based system, all that was required was the addition of an I^{2}C hardware block to the Microblaze's peripheral data bus and a simple reprogramming of the configuration PROM.

In addition to the flexibility the FPGA provided in the data channel, this new revision of hardware integrated previously external peripherals onto a single PCB. Among the peripherals were a barometric pressure sensor paired with a high-resolution analog-to-digital converter (ADC), on-

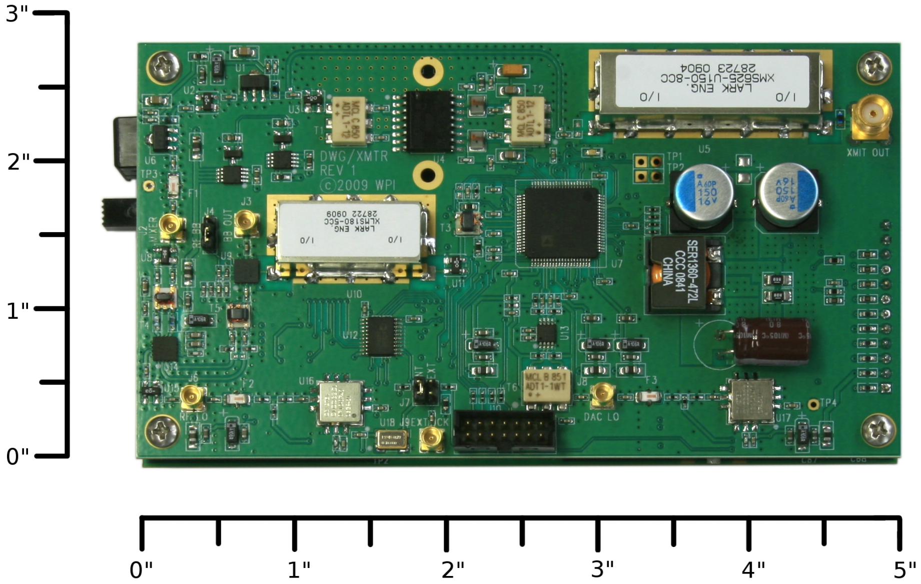


Figure 2.6: Revision 2 DWG/XMTR Board

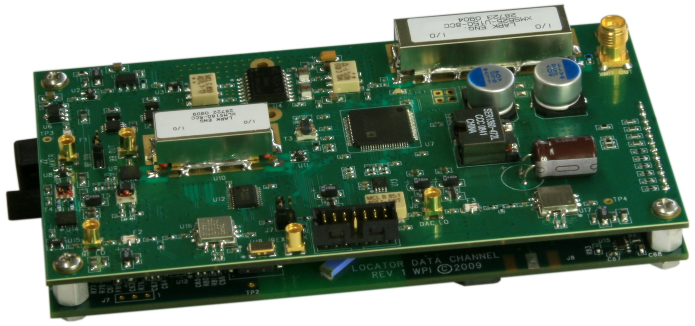


Figure 2.7: Revision 2 PCB Stack (RF Hardware on Top, Data Channel Underneath)

board nonvolatile storage, and an additional WPI-designed radio daughter board for short-range communications with physiological sensors. These various sensors can be seen in the annotated data channel diagram (Figure 2.8).

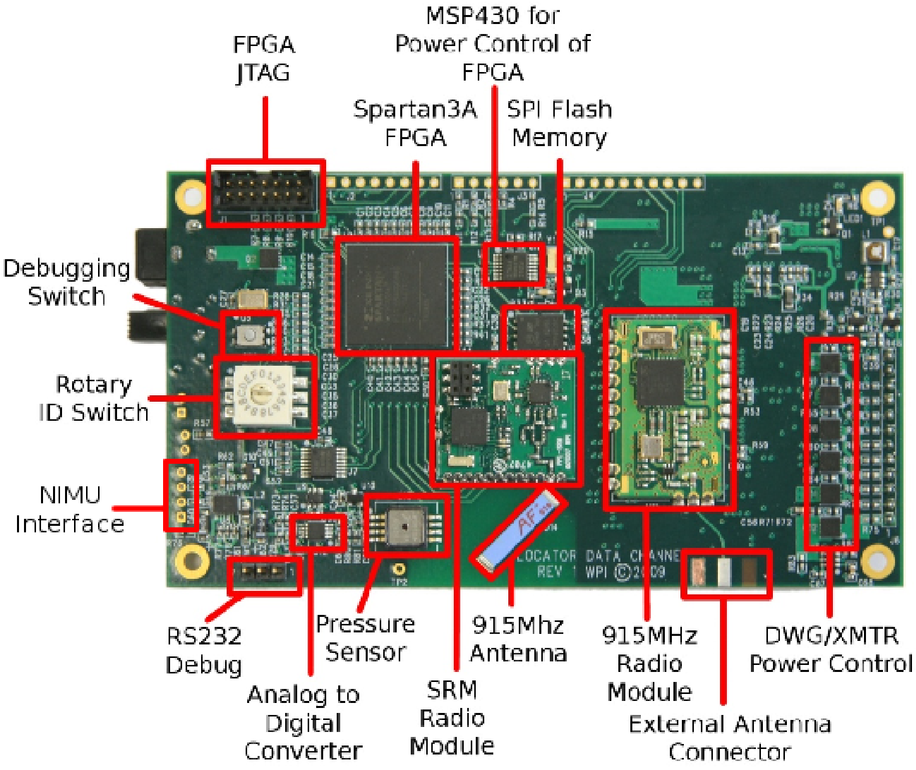


Figure 2.8: Annotated Revision 2 Data Channel

The nIMU interface required a separate supply voltage used nowhere else in the system. A special header was added to supply this voltage from a dedicated regulator on the PCB, along with pins for communication and ground. The header provided a simple plug-and-play interface for the nIMU. Data was captured via the interface and retained in the nonvolatile storage for later processing. This convenient configuration also allowed for concurrent sampling of various other sensors in the system from which a fusion-based approach could be used for location estimation.

The revision 2 locator was used from 2009 though 2010 and proved to be a successful revision. Although not without its faults, the goal of true versatility was proven time and again as unforeseen tasks were executed on this board with ease and efficiency not seen in previous revisions. Several new designs added to this board were carried over to the next revision and the limitations such as peripheral interfacing and RF hardware issues were addressed.

Transceiver Boxes

The transceiver boxes in revision 2 represented a large step forward for the reference hardware of the PPL system. In revision 1, the reference stations consisted of little more than analog RF circuitry that conditioned the received signal for it to be sent to the base station over 50m baseband cables. The data acquisition hardware for the revision 1 system was located at the base station and connected directly to the processing computer. Revision 2 featured stand-alone FPGA-based reference units capable of full transceiver functionality. Figure 2.9 shows a transceiver box with the cover removed, exposing the PCB.

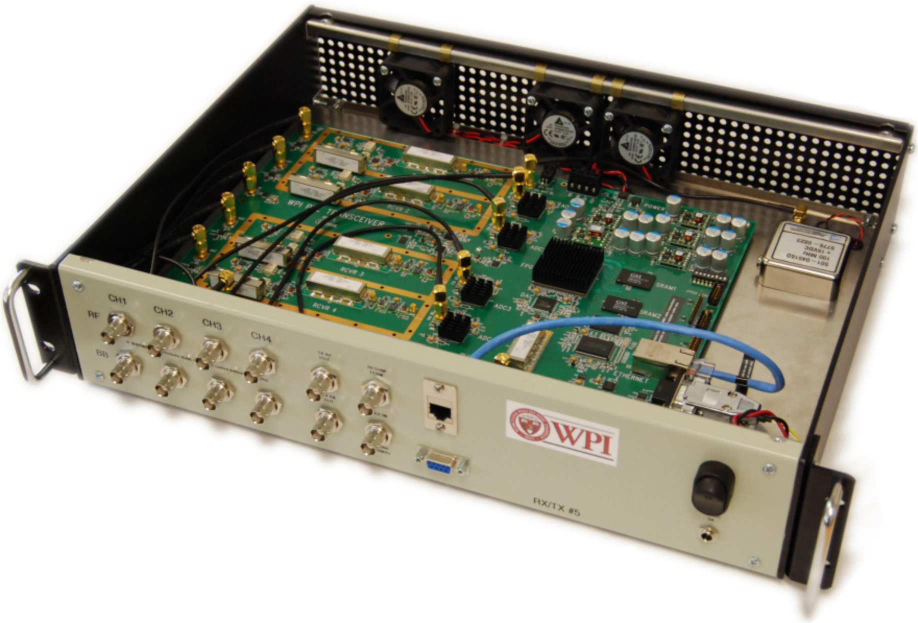


Figure 2.9: Revision 2 Reference Hardware – The Transceiver Box

The transceiver box was designed to be a stand-alone data acquisition and transmission unit to suit the needs of the PPL project. Figure 2.10 shows a system-level block diagram of this hardware. Four high-speed (400MHz) ADCs are included in this design to allow for simultaneous sampling of all four antennas. High-speed SRAM is used for ADC sample storage. In addition to the receiving circuitry, a single transmit channel is included that is multiplexed, allowing transmission out of any one of the four antennas at a time. Only one transmit channel is included because there is no functional reason for more than one antenna in the PPL system to be transmitting at a time.

A gigabit Ethernet interface is also included to facilitate moving sampled data off the transceiver as well as providing a channel for a control protocol. The protocol used is a PPL-developed protocol named the Control and Messaging Protocol (CAMP). CAMP is responsible for system-level control such as starting data captures, system configuration, and data retrieval.

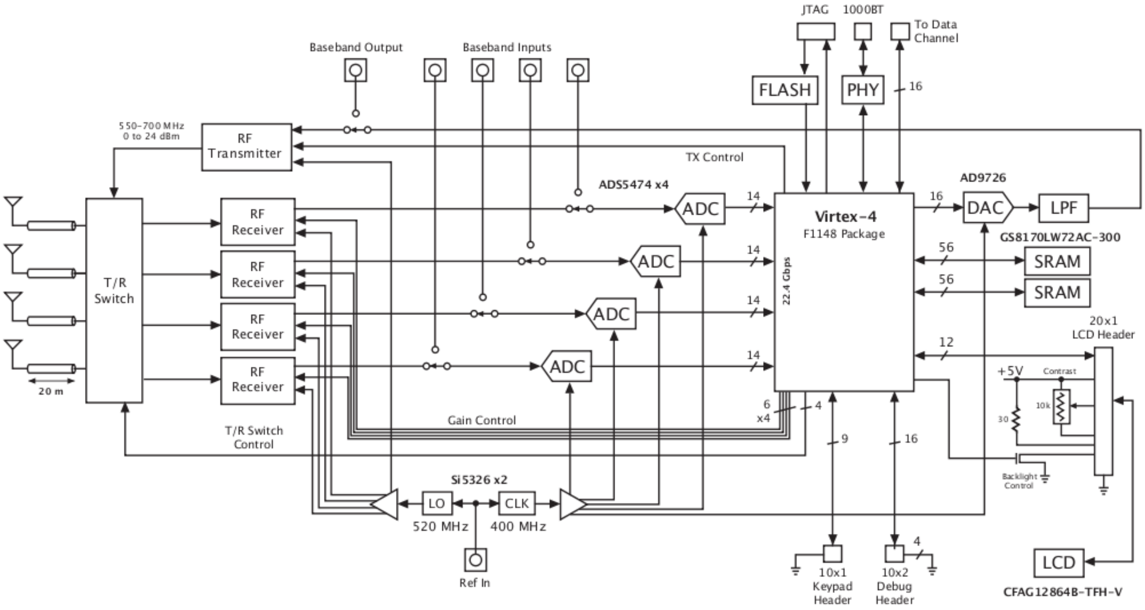


Figure 2.10: Transceiver Box Block Diagram

At the core of the transceiver is a Xilinx Virtex 4 FPGA. This high-speed device acts as the interface that integrates the various subsystems together. With this high-speed FPGA device, control and data paths for the ADCs, transmit channel, Ethernet, and SRAM can be implemented in VHDL. In addition to these custom hardware interfaces, Xilinx's softcore 32-bit microcontroller, the Microblaze, is also implemented on the Virtex 4. The Microblaze provides the ability to control the system through custom-written software. This allows for quicker and easier changes to system functionality since software changes require less effort than hardware design changes.

Although originally designed with the σART algorithm in mind, the transceiver box provided a platform for initial proof-of-concept TART work. With the transceiver's ability to both transmit and receive PPL's multicarrier ranging signals, it was capable of simulating the role of a mobile TART locator. From this testing, the real-world viability of TART was shown. However, the transceiver's large size and lack of battery power quickly became a hindrance during TART testing. The need for a mobile transceiver locator that supports the TART method became a goal for the PPL project at this point.

2.2 TART Location Method Overview

The driving force behind the revision 3 of the locator was the implementation of a mobile transceiver capable of supporting the PPL project's newest innovation in indoor location: Transactional Array Reconciliation Tomography (TART). This is the state of the art in RF indoor location developed at WPI. TART is intended to replace the previous location algorithm, σART. σART is described below to provide the necessary background for understanding PPL's multicarrier indoor location techniques.

2.2.1 σART

The previous method of location estimation, σART, was essentially a time difference of arrival (TDOA) based method[9][3][10]. This method was based around the use of a mobile transmitter and statically placed receiver reference stations. The basic system architecture can be seen in Figure 2.11.

In this system, the mobile transmitter emits a multicarrier signal, which is then received by the statically placed receiver antenna arrays. All antennas are sampled in a sufficiently small window of time on the order of tens of milliseconds. The received data is then processed with the σART algorithm at the base station on a computer.

The data fed into σART has a rephasing method applied to it. This is an exhaustive scan over a spatial grid that seeks to maximize the σART metric. This method creates a discrete spatial scan grid that is assumed to contain the transmitter. For real-world applications, this usually is an area that is bounded by the receive antenna arrays. Each point in the grid has the σART algorithm applied to it. σART produces a metric that indicates the likelihood that the transmitted signal originated from

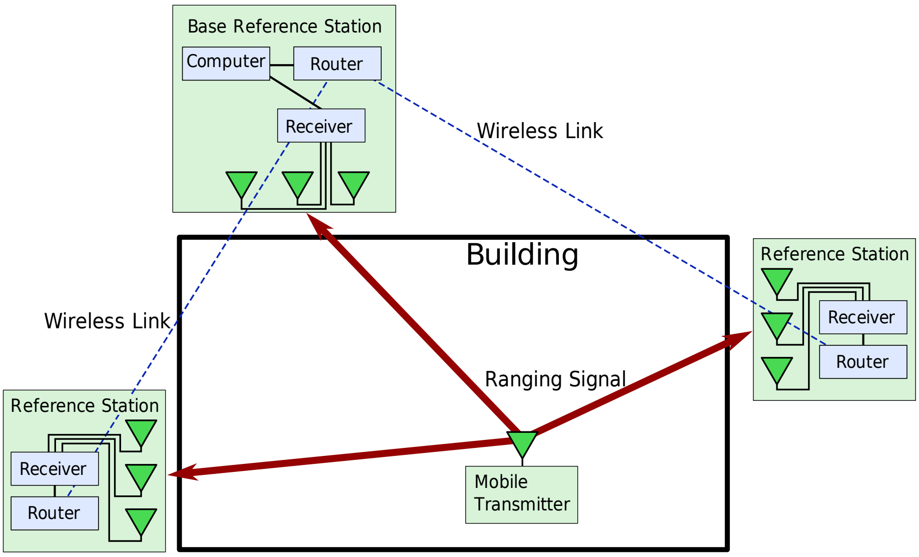


Figure 2.11: σART System Architecture[3]

that point in the scan grid.

The process of scanning the spatial grid through rephasing is shown in Figure 2.12. The left of this figure shows several receive antennas, represented as red circles, as well a transmitter device, represented by the blue square. The time delays from the transmit location to each antenna is shown as $t_{0,n}$. On the right of the figure, a rephasing scan grid is imposed on top of the antenna locations with the grid representing the locations where the σART metric is evaluated. For each of the grid locations, a precomputed signal is produced by creating a signal that corresponds to the time delays, shown as $\#t_{0,n}$, associated with a signal originating from that grid location.

At each grid location, a metric is produced based on the strength of a correlation between the true received signal and the precomputed rephasing signal for that grid location. The right of the figure shows the process of correlating the true received signal (green lines) and the grid location two units to the right (red lines). The stronger the correlation between these two signals, the higher the σART metric will be, and the greater the likelihood that that scan grid location is the true originating location of the transmission. With this process completed over every location in the scan grid, the metric value maximum represents the true location of the transmission in a successful scan.

Figure 2.13 shows an ideal simulation of the σART method. A perfect RF channel is assumed

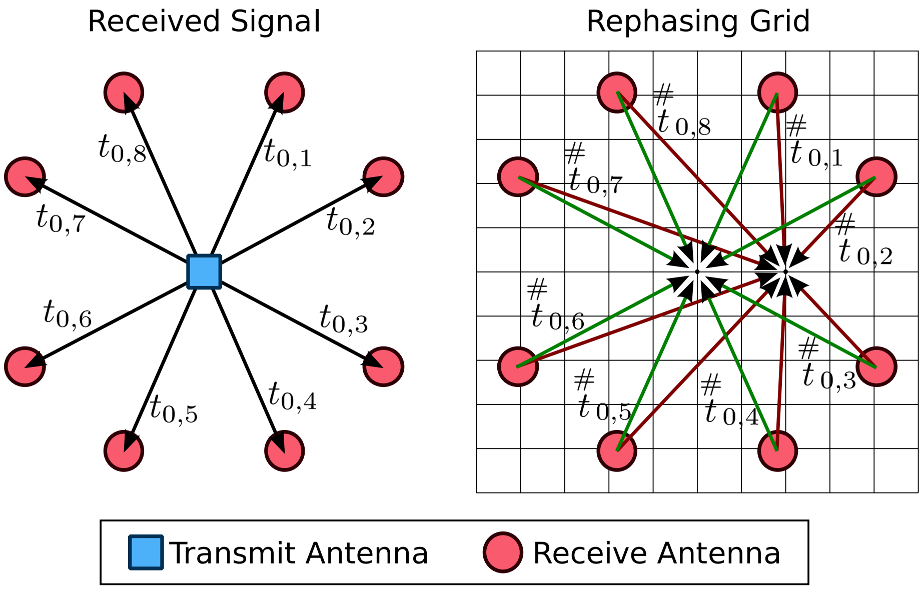


Figure 2.12: Rephasing Procedure[3]

with no interference or multipath. The color in the figure represents the strength of the metric at a given scan point, ranging from deep blue for an unlikely location to deep red for a very likely location. The transmitter is represented by the black X, and the circular marks represent 16 receive antennas. All subsequent scan plots of σART and TART will follow this graphical convention. As can be seen, in an ideal simulation a very sharp peak of the metric forms around the transmitter location and zero error is achieved.

In ideal simulations, σART performs excellently in almost all reasonable configurations of receive and transmit geometry. When more real-world factors are accounted for in simulations, σART 's shortcomings become apparent. One factor that is commonly seen in all real-world situations is multipath.

Multipath

Figure 2.14 shows an explanation of multipath signals. When a signal is transmitted and subsequently received, the shortest distance between the two points (and thus the shortest time delay) is referred to as the direct path. In addition to the direct path signal, reflections of the original signal are also received. In Figure 2.14, the green triangles represent RF reflectors. When the transmitted

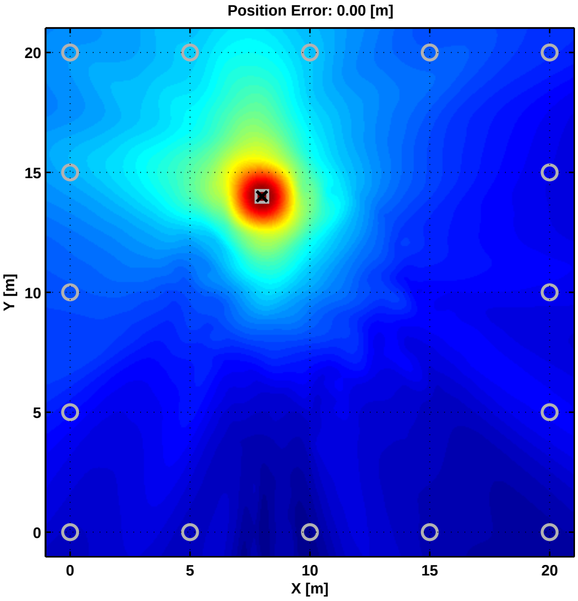


Figure 2.13: Ideal σART Simulation[3]

signal hits these reflectors, it is redirected and can end up at the receiver. These signals will appear as a time-delayed version of the direct path signal.

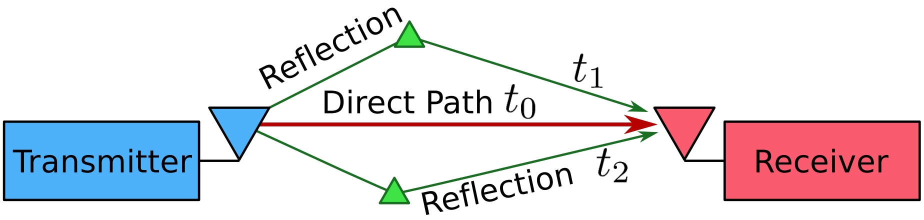


Figure 2.14: Multipath[3]

The addition of these multipath signals can cause ambiguity in σART 's output metric. However, if the direct path signal is sufficiently stronger than the multipath reflections, σART will often be able to still correctly identify the correct location.

Figure 2.15 shows another σART simulation with a single ideal reflector added. Because σART is a TDOA based method, the reflector simply appears as another transmitter and the difference in delay times is unknown. With an ideal reflector, both the true transmitter and the reflector will produce the same metric intensity. In this case, σART incorrectly chooses the reflector as the location, resulting in gross position error. After the explanation of the TART method, below, real-world results can be found for σART in a comparison of σART versus TART.

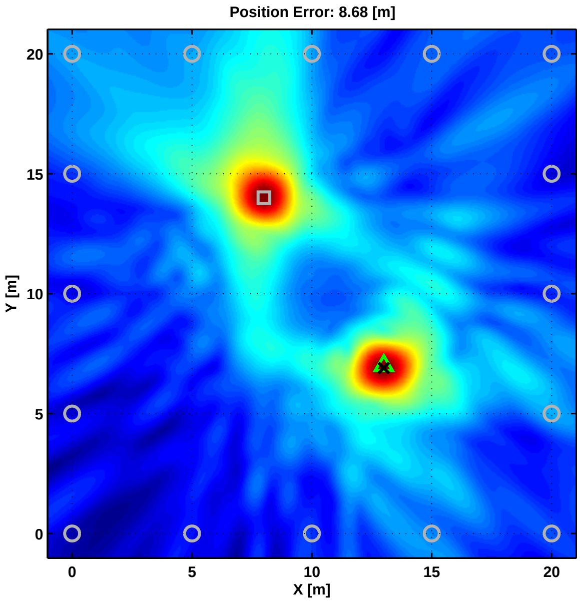


Figure 2.15: σART Simulation With Single Ideal Reflector[3]

2.2.2 TART

The next incarnation of the PPL location algorithm is TART (Transactional Array Reconciliation Tomography)[3]. TART is in ways an extension of σART, but it differs in some key fundamental aspects.

A system architecture diagram of TART is shown in Figure 2.16. This architecture is similar to the σART setup shown in Figure 2.11. The major difference in the TART system architecture is that, rather than a unidirectional signal transmitted from a mobile locator, a bidirectional transaction is carried out between the reference stations and a mobile transceiver.

An ideal simulation of both σART and TART can be seen in Figure 2.17. In this simulation, the same ideal data was processed with σART and then TART. Both methods produce zero error for this ideal case; however, some of the inherent differences between the two methods can be seen in this

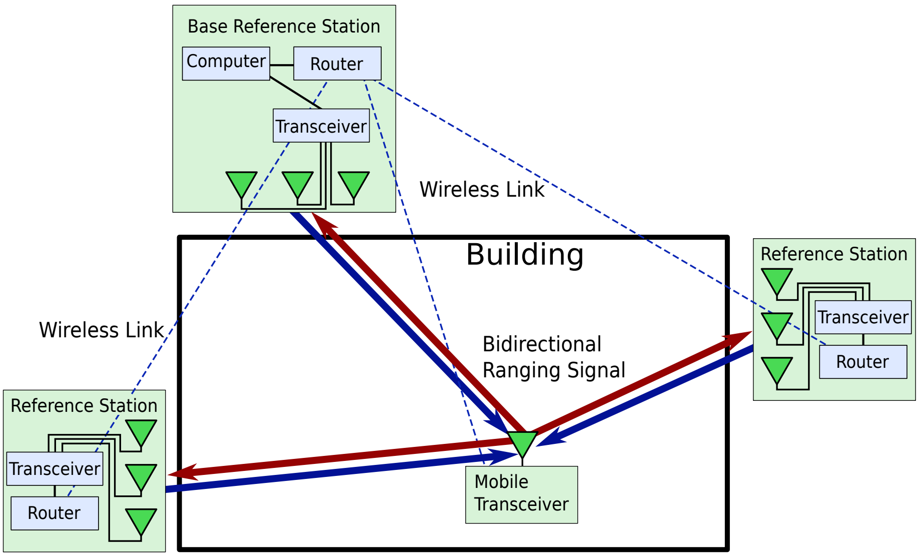
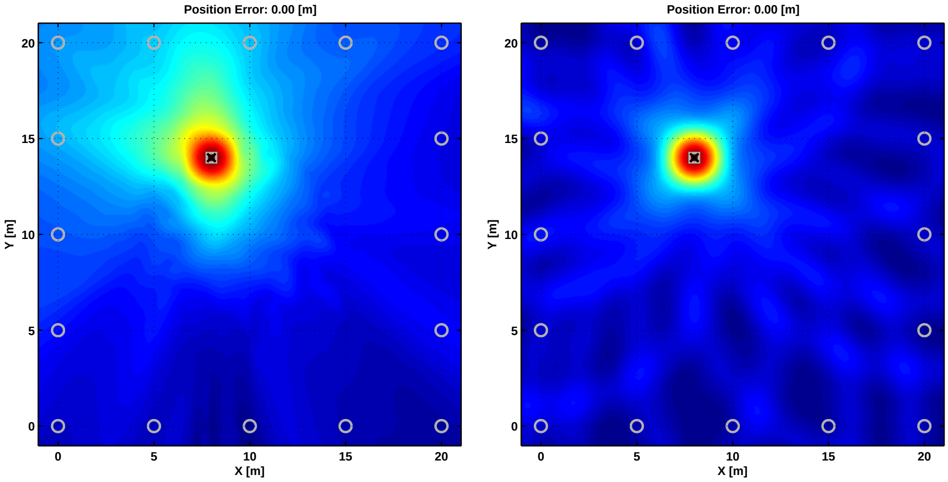


Figure 2.16: TART System Architecture[3]

comparison. The σART scan shows a mostly constant ramp of the metric value as the true location is approached. In the TART scan, a series of circles appears around each reference antenna, and the truth location has a tight peak in the metric where the circles intersect. This effect is due to the TOA nature of TART and is one of the keys to TART's improvement over σART.

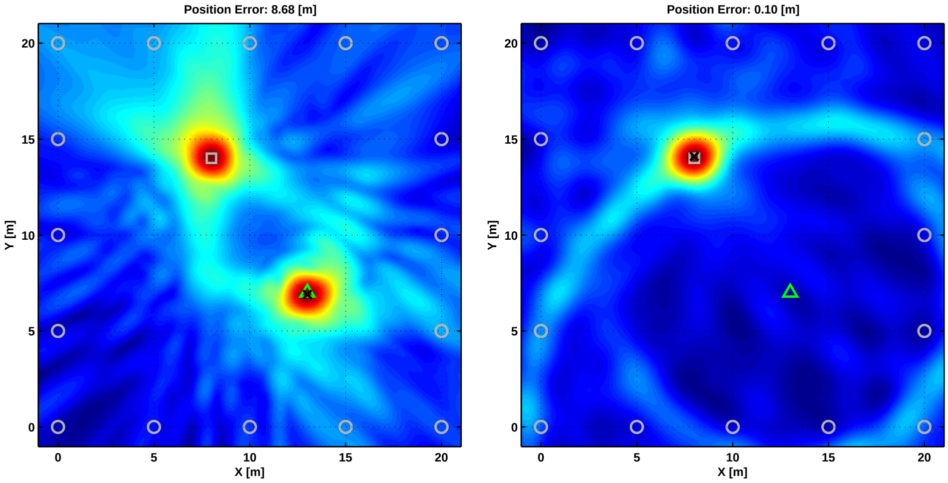
σART is a TDOA-based method that can result in ambiguity of location when low direct path and high multipath are present. TART is a TOA method that mitigates many of these problems. Figure 2.18 shows a simulation with a single ideal reflector represented by the green triangle. The σART image shows serious ambiguity as to the true location, and the algorithm incorrectly chooses the reflector as the truth location, resulting in gross error. The TART image shows a zero error. Not only did this simulation produce zero error, the reflector did not show an increase in the metric at all. This effect is a result of TART's TOA nature.

A reduced geometry reference antenna configuration is another situation in which TART significantly outperforms σART. Both σART and TART benefit from a larger antenna count and geometrically diverse configuration. The number of antennas, as well as their arrangement, has a serious impact on system performance. However, TART copes with reduced geometry far better then does σART.



(a) σART (b) TART

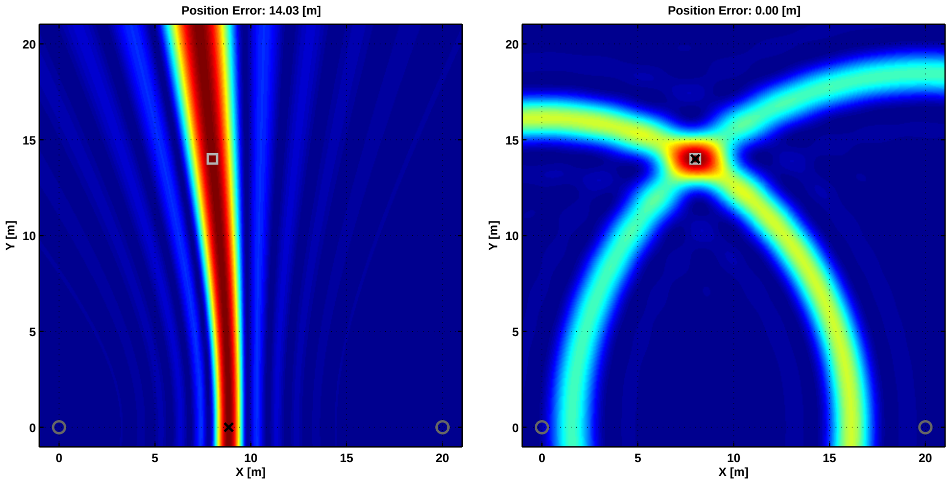
Figure 2.17: σART vs. TART - Ideal Simulation[3]



(a) σART (b) TART

Figure 2.18: σART vs. TART - Single Reflector Simulation[3]

Figure 2.19 shows an ideal simulation with two reference antennas for both σART and TART. As can be seen in the figure, σART produces an elongated peak of the metric, which introduces ambiguity as to the true location. Not surprisingly, σART chooses an incorrect location along this maxima ridge, resulting in gross error. On the contrary, TART produces a tight peak at the true location and results in a zero error for this simulation.


Position Error: 14.03 [m]

Position Error: 0.00 [m]

(a) σART

(b) TART

Figure 2.19: σART vs. TART - Two-Receiver Simulation[3]

This reduced geometry simulation helps to illustrate the key difference between the σART and the TART methods. With TART's TOA-based approach, an additional piece of timing information is obtained that σART does not have available. In the TART simulation, two rings of increased metric can be seen encircling each of the two antennas. These can be thought of as each antenna's contribution to the final TART metric.

As with any TOA ranging method, each antenna essentially produces a one-dimensional range estimate. This allows for an accurate location estimate with only two antennas assuming a properly constrained scan grid. The TART method will produce two maxima in an infinite scan grid because the range metric circles will intersect at two locations.

σART is missing the extra piece of timing information that TART makes use of. Because of this, it would take at least three antennas to produce a viable location estimate with σART. As can be seen in the simulation, σART can produce only a ridge on which the location lies. Although the true location does lie on this ridge, σART lacks the proper information to pinpoint the location.

TART Transactions

At the core of TART's TOA method is a two-way TART transaction. Previously with σART, the mobile transmitter was turned on for a time window on the order of a few milliseconds. This window was known to all units within the system by microsecond-level timing synchronization achieved through use of the digital radio interface. Within this known transmit window, all reference stations sampled the receive antennas, and the results were processed with the σART algorithm.

The TART algorithm operates with a new transactional method, shown in Figure 2.20. Rather than a single transmission, TART transactions consist of two distinct transmissions between the reference antennas and the mobile locator. The order of these two transactions is not crucial to the operation of TART. After both units have transmitted, the received signal data must be moved to the base station for processing. The bottom two transmissions represent the reference and mobile units each sending their received signal data to the base station.

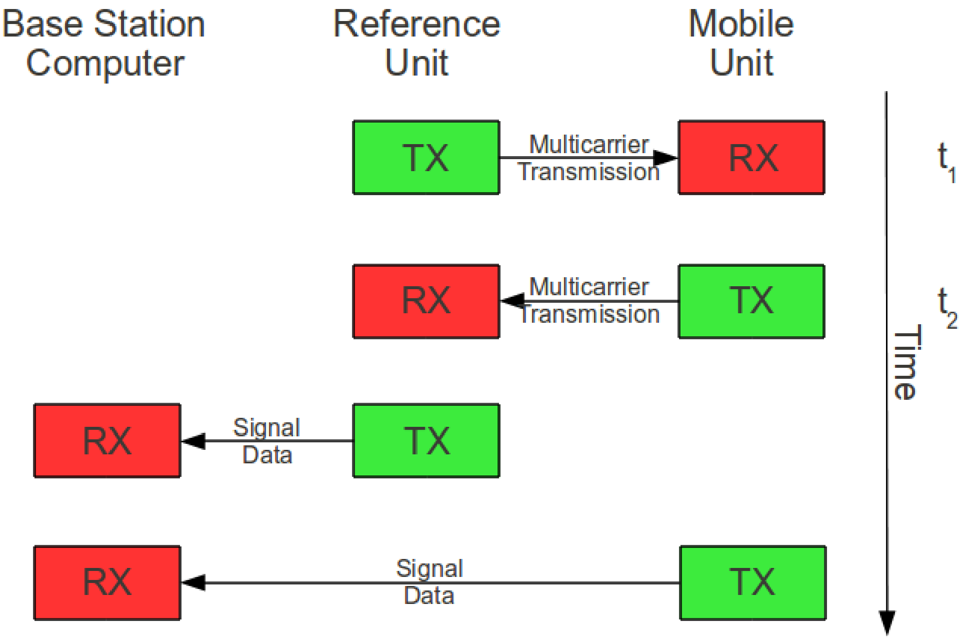


Figure 2.20: Anatomy of a TART Transaction

The σART algorithm is able to deal with an unknown global offset in the transmitted signal due to its TDOA-based approach. TART requires this offset to be zero to function. In order to cope with this need, the data from the two way transaction is used to solve for this global time offset.

Clock Drift Prediction

In the real world, there is one other factor that needs to be accounted for to achieve the timing synchronization TART requires. In initial testing of TART, highly stable clock sources, such as atomic clocks, were used. In a mobile locator implementation, atomic clocks cannot be used due to size and power constraints. The best available clocking resource that is appropriate for use in mobile locator hardware is a crystal oscillator. With inherently unstable crystal oscillators used in the system, timing synchronization is lost due to clock drift over the time of the TART transaction.

To combat the drift of inexpensive crystal oscillators, a third transmission is added to the TART transaction. This transmission is used to estimate the clock drift over the time of the transaction. Figure 2.21 shows the TART transaction with this added transmission, the first reference unit to mobile unit transmission.

Using the transmissions from the reference unit to the locator at time t_0 and t_1, linear interpolation can be used to achieve a low error estimate of clock drift at time t_2. Figure 2.22 shows this process. The red line, $\tilde{\tau}(t)$, represents the true clock drift over time. The two transmissions at times t_0 and t_1 are used for the linear clock prediction. The third transmission, t_2, represents the transmission where clock drift must be accounted for. At this point, the difference between the true clock drift and the predicted clock drift is the error represented by ϵ. Previous research on the PPL project showed that this error will remain sufficiently small over the time of a TART transaction to produce viable location estimates. The drift that occurs over the time of a TART transaction with crystal oscillators, would introduce error in the solution of less then 0.5ns 99.7% of the time.[3]

Consider Equation 2.1, which shows the distance traveled by an RF signal in free space over 1ns. This value, 0.3m (approximately 1ft), represents the amount of location estimation error that will be introduced due to every nanosecond of synchronization error. With the total timing error introduced into the system due to clock drift reaming below 0.5ns, the position error would only be 0.15m. This amount of error falls within the goals of the PPL project's requirements.

$$\frac{299,792,458\text{m/s}}{1,000,000,000\text{ns}} \approx 0.3\text{m} \tag{2.1}$$

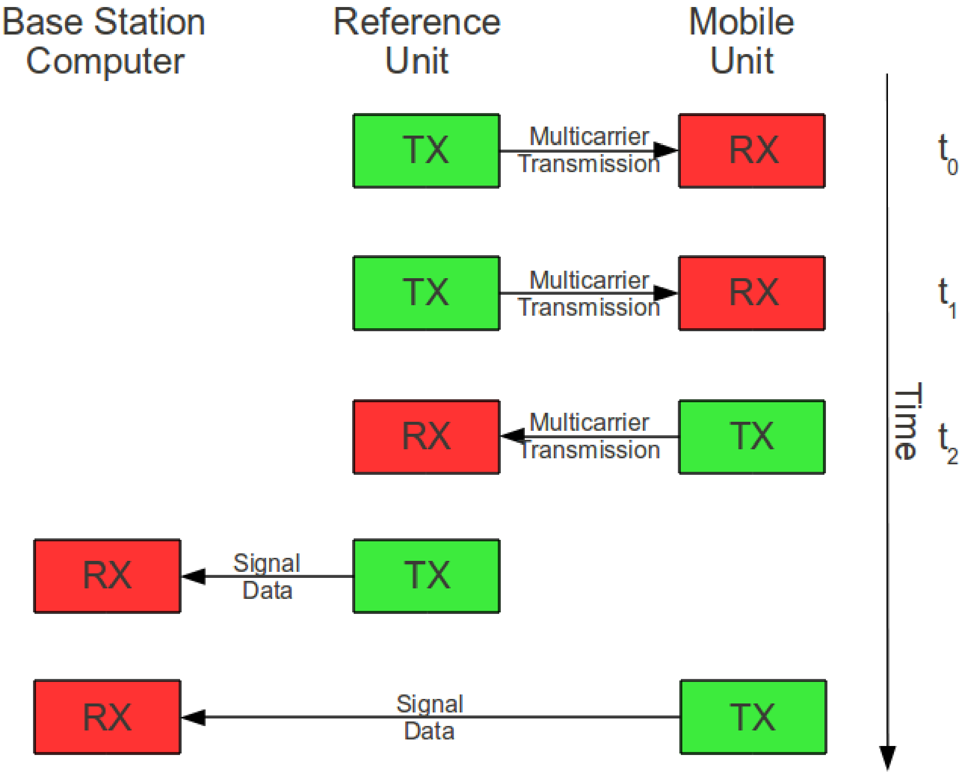


Figure 2.21: Anatomy of a TART Transaction - Linear Clock Prediction

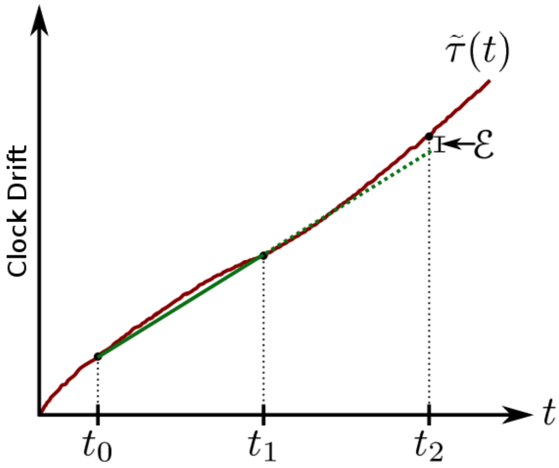


Figure 2.22: Linear Clock Drift Prediction[3]

Marginally better prediction could be had by using t_0 and t_2 for the linear clock drift prediction; however, there are hardware considerations that make this complicated. Figure 2.23 shows this alternative approach with the reference unit transmitting at times t_0 and t_2.

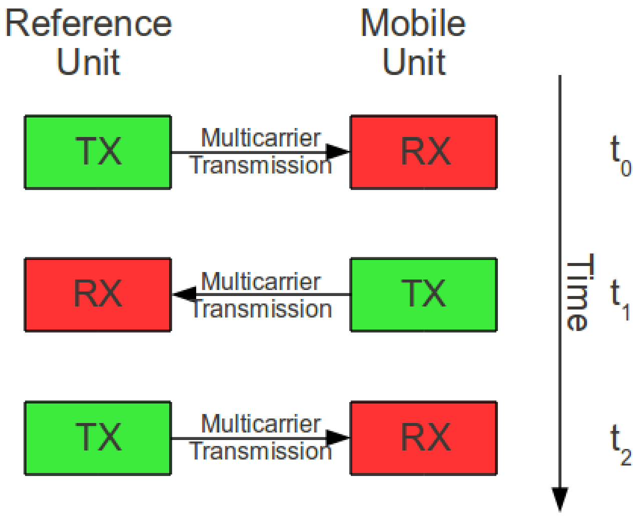

Reference Unit
Mobile Unit
TX
Multicarrier Transmission
RX
t0
RX
Multicarrier Transmission
TX
Time
t1
TX
Multicarrier Transmission
RX
t2

Figure 2.23: TART Transaction - Alternative Clock Drift Prediction Scheme

Switching hardware from a transmit state to a receive state takes a finite amount of time that is not negligible to TART. With the transaction run as shown in Figure 2.21, only a single transmit/receive transition is needed between the second and third transmissions. Any gains that might be had from a more accurate clock drift prediction would be minimized or lost due to the extra time needed for a second transmit/receive transition. Additionally, the extra transition would increase the complexity of the transaction from an implementation standpoint.

2.2.3 Real-World Results

Although TART consistently outperforms σART in a wide array of simulations, as shown above, it is important to see these methods compared in real-world situations before investing in hardware development. Several tests have been conducted with both the σART and TART methods at the same locations. One such test at WPI's religious center was run with a unique ladder-mounted antenna setup, shown in Figure 2.24, that consisted of 16 antennas located on a single side of the building. This test helps illustrate the advantages TART can give in the real world.

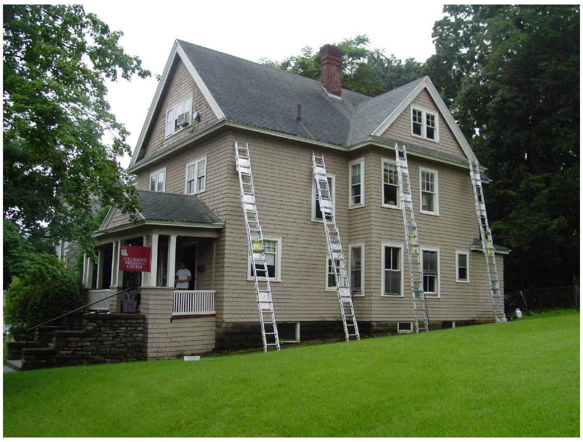


Figure 2.24: WPI Religious Center TART vs. σART Test Setup

In this test, data was collected at a total of 11 different points throughout the first floor of the three-story residential structure. The test data was analyzed with both the σART and TART algorithms over a three-dimensional scan grid. Figure 2.25 shows the horizontal error vectors for the first floor, with TART clearly showing better accuracy at nearly every location.

A three-dimensional scan at location 1 is shown in Figure 2.26. As expected with a coplanar antenna configuration, σART shows a large amount of ambiguity in the x direction. On the contrary, TART's metric shows a far more contained metric peak, which results in a far better estimation of the location.

Additionally, Table 2.1 shows the overall error statistics for σART and TART for the test on the first floor of the religious center. It is clear that TART significantly outperforms σART, especially in XY error. The Z errors for σART and TART show no clear winner; however, both methods produced acceptable overall Z error results.

This test demonstrated that TART has viability to provide increased system performance in a real-world environment. Not only does TART outperform σART, it excels in this reduced antenna geometry setup. The results from this test show that pursuing a new hardware platform to support the TART location method is the next logical step for the PPL project.

Further detail and more in-depth explanation of the inner workings of σART as well as TART and additional real-world test results can be found in a previous thesis and papers produced by the

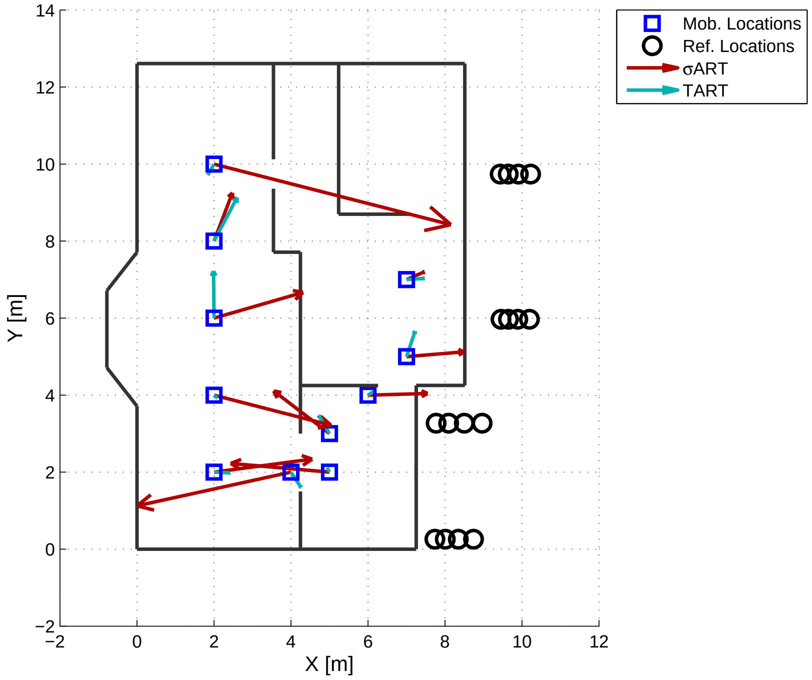
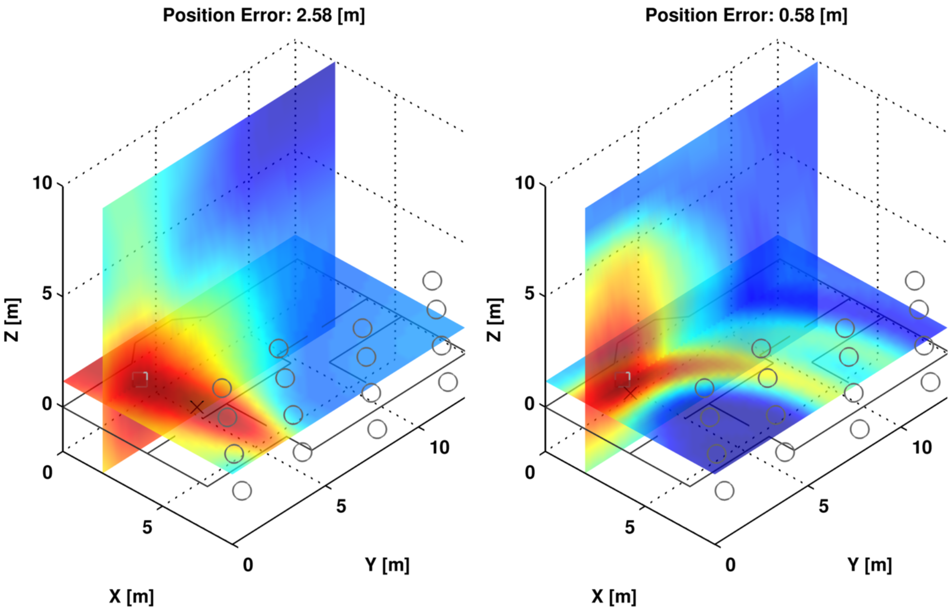


Figure 2.25: Residential Building Test: 3D, First Floor Error Plot[3]



(a) σART

(b) TART

Figure 2.26: σART vs. TART - Residential Building Test: 3D, Location 1[3]

Error [m]	σART	TART
RMS	3.02	0.81
Med.	2.41	0.75
Max.	6.52	1.50
XY RMS	2.95	0.65
XY Med.	2.41	0.47
XY Max.	6.35	1.28
Z RMS	0.62	0.48
Z Med.	0.32	0.39
Z Max.	1.49	0.77

Table 2.1: Error Summary - Residential Building Test

PPL project[3][9].

With a solid history of the PPL hardware revisions and an explanation of the σART and TART algorithms, the requirements and design of the next revision hardware can be considered. The revision 3 hardware, the focus of this thesis, fills the need for a mobile hardware implementation that supports the TART location algorithm. Full transceiver capabilities along with many other evolutionary changes derived from previous revisions of the hardware examined in this chapter weighed heavily in the design process. Additionally, with the new UHA system architecture, several design requirements need to be laid out such that the new hardware can fulfill the hardware needs of the entire PPL system. Chapter 3 covers the requirements and design phase of the revision 3 hardware, as well as component selection and justification.

Chapter 3

Locator Requirements and Preliminary Design

The redesign of the new locator hardware was in some ways an evolutionary design based on the previous revisions of locators. However, the addition of support for the TART method of location stipulated several radical changes in the fundamental system design. The largest consideration for TART support was the addition of a receive channel to the locator hardware allowing for transceiver capabilities. In addition to the support of TART, a new system architecture revolving entirely around the revision 3 hardware was devised. With this new system architecture, the revision 3 hardware will replace all previous hardware in the PPL project. Through understanding of the new system architecture and the role the revision 3 hardware has within it, requirements for the new hardware began to be defined.

3.1 Unified Hardware Approach

The revision 2 system architecture consisted of two distinct hardware components: the mobile locator and the reference transceiver box. One goal of the revision 3 hardware was to replace these two separate hardware devices with a single device that could function in different modes to satisfy all the hardware needs of the PPL project. This radical new approach to the PPL hardware organization is referred to as the Unified Hardware Approach (UHA). The main premise of this approach is to replace the four transceiver boxes with up to 16 pieces of the new locator hardware. The PPL system has been based around a 16 antenna approach, with arrays of four antennas connected to each transceiver box. The new locator hardware, with its single transceiver channel, is able to support only one antenna, necessitating 16 devices throughout the system. Although the overhead of setting up and maintaining 16 devices is a cumbersome task, the advantages described in this section far outweigh this cost.

Figure 3.1 shows the system architecture of the revision 2 PPL system. The configuration included large rack-mounted setups with the revision 2 transceiver boxes. Arrays of four separate antennas were hardwired to each of the four transceiver boxes. In addition, gas-powered generators were needed to power the transceiver setup, atomic clocks, and WiFi data link solution when testing was performed in an outdoor location, which is the common practice.

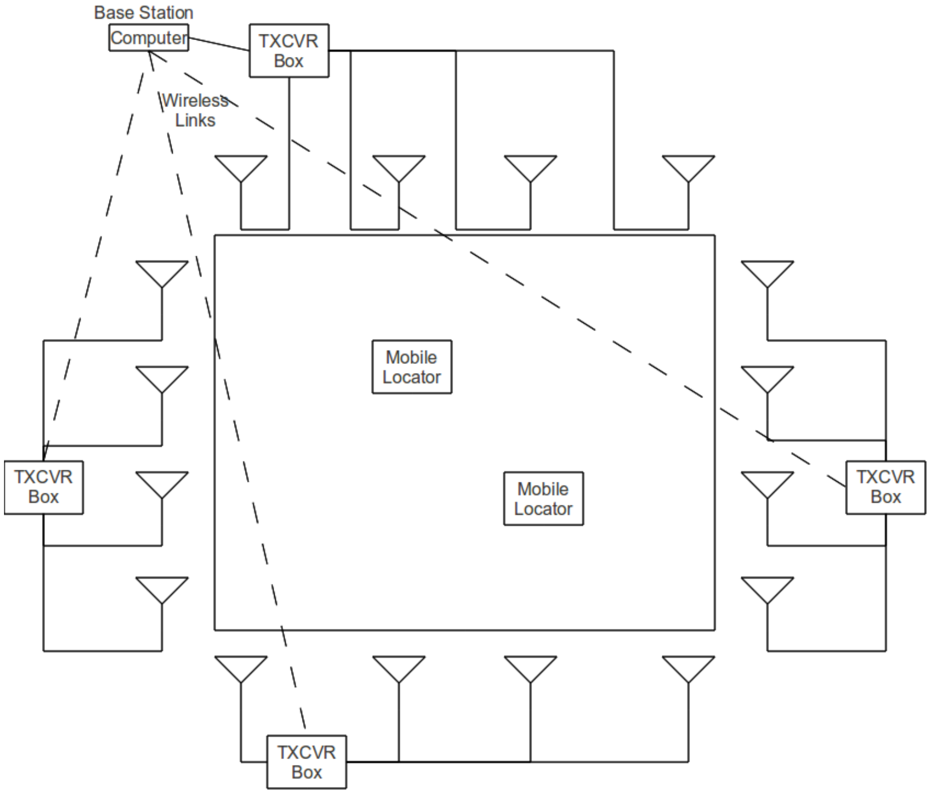


Figure 3.1: Revision 2 System Architecture

In the new system configuration (Figure 3.2), the transceivers are replaced with discrete battery-powered hardware units at each antenna. To replace the functionality of the revision 2 transceiver box, each unit will need transmit and receive capabilities, high-speed wireless communications, an Ethernet interface, and high-speed ADC sample storage RAM. The data link is handled by the on-board high-speed data link or possibly a WiFi fallback device using the Ethernet interface. Each of these discrete reference stations will fulfill the PPL project's need for RF signal acquisition and transmission, as well as transporting the captured data to the base station computer via wireless link.

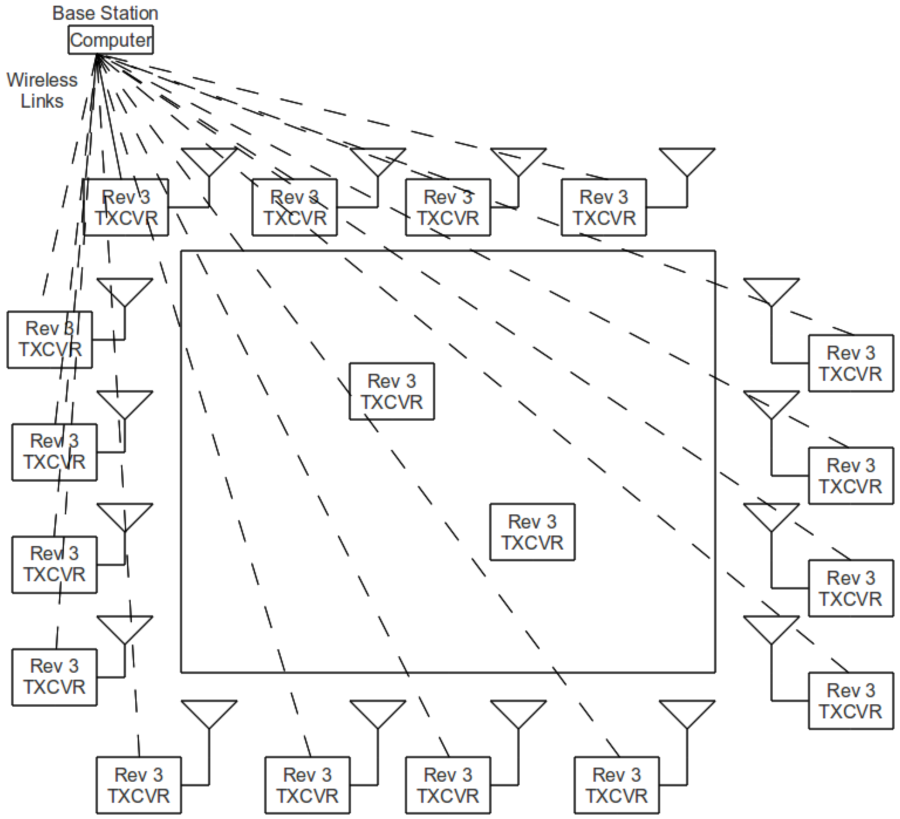


Figure 3.2: New Hardware TART System Architecture

The mobile locator devices for the UHA system architecture need to fulfill all the functionality of previous locator revisions as well as provide receive capabilities to support the new TART location method. These devices need to be personnel mounted, support transmit and receive capabilities, have a high-speed wireless data link for communication and data transfer, maintain compatibility with previously used physiological sensors, contain the various peripheral sensors used by the PPL project, and provide flexibility to support future applications within the PPL project.

When the viability of this approach was considered, the pros and cons had to be weighed. Although there was a lot to be gained from the UHA, there were some logistical issues that had to be overcome. One of the first major advantages of UHA is the simplified system setup and appearance. The old system had four large, heavy, rack-mounted setups that required gas-powered generators to run. Additionally, there was the need for long RF cables to connect the antennas. The cables are cumbersome to work with and also introduced almost 6dB of attenuation before the RF signal even reached the hardware. In UHA, each reference station is a discrete setup consisting of battery-powered revision 3 hardware, an RF antenna with a short RF cable (less then 1m), a stand, and possibly WiFi fallback data link. Preassembled antenna stands would not only allow for quicker system setup for testing but also give more flexibility in test geometry as well as a clean and practical visual look.

From the new UHA system architecture, a piece of hardware was developed to support all its needs: the revision 3 hardware. A system-level block diagram of this device is shown in figure 3.3.

A large amount of functionality in this new system level design is based on the revision 2 transceiver box and previous locator devices. Key among these is the transmit and receive capabilities. Selection of the silicon device for each of these channels, DAC for transmit channel and ADC for the receive channel, as well as the design of the associated analog systems will pull heavily from experience gained from previous design efforts. Another key component is the selection of a high speed data link solution that can fulfill the needs of both the reference stations and locator applications of the new hardware. In addition to these major subsystems of the new design are many smaller peripheral components, including sensors and interfaces used in the PPL system.

3.2 High Speed Data Link

When the TART method was implemented with new hardware, the issue of sending the TART ranging signal data back to the base station became an issue. These signals are sent to the base station via digital data link and require far more bandwidth compared with data sent in previous revisions. In order to support the higher data bandwidth needs associated with TART, an improvement had to be made on the 915MHz module used in previous designs. The 915MHz Xemics XE1205–

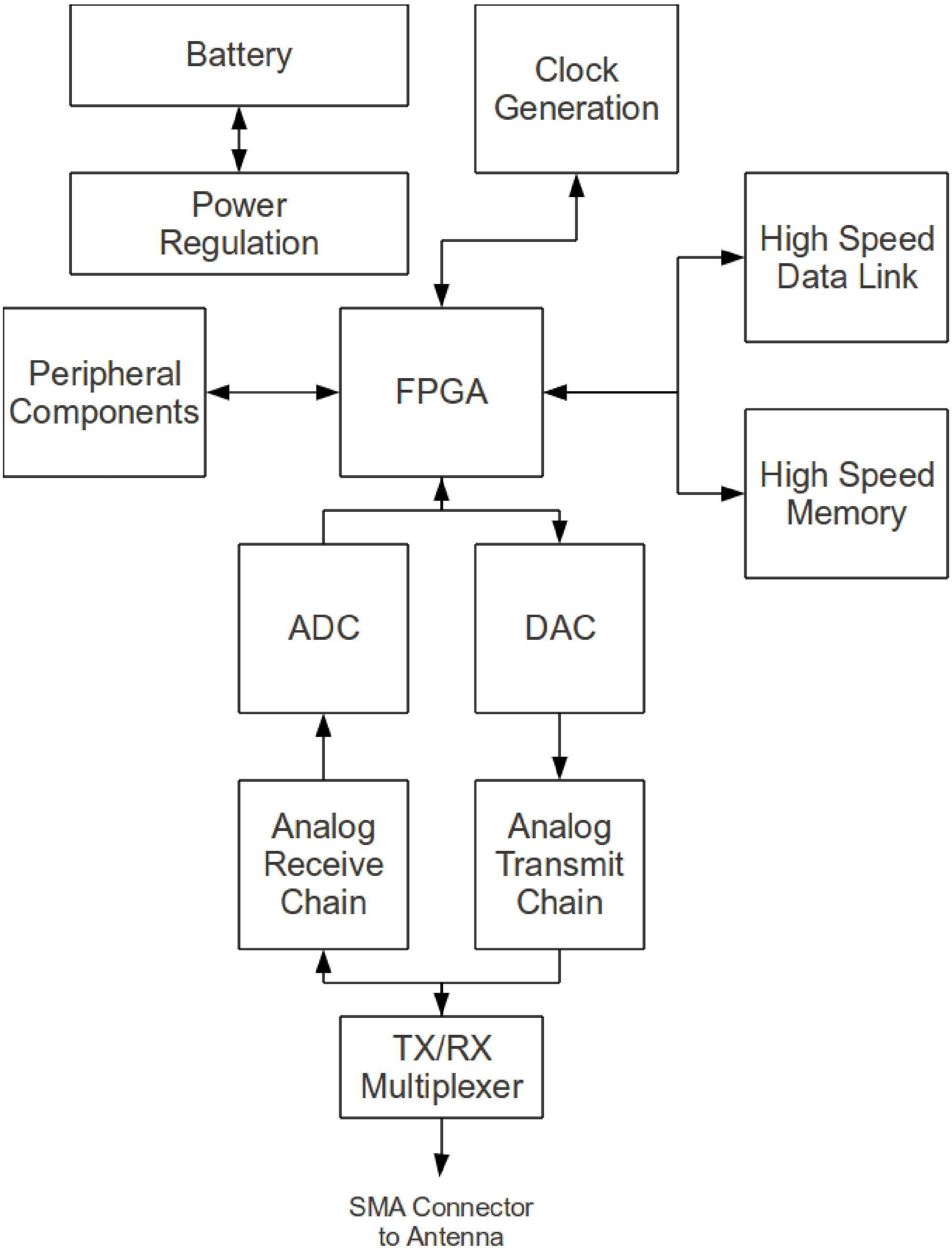


Figure 3.3: Revision 3 System Level Block Diagram

based module was a great solution in previous designs. It provided simple interfacing, low-level control, low power consumption, and good range coverage. With previous designs in which data throughput was not an issue, the Xemics module was used at a baud rate of 19200, which was able to penetrate multi-floor buildings. Even ignoring the loss in range that would come with increasing the data rate, the maximum rate for the Xemics (304.7kbits/sec) still would be inadequate for multiple locators running TART[11].

3.2.1 Throughput Requirements

Before specific data link solutions were considered, the exact data link requirements had to be defined. These requirements scale with the number of desired locators in the system, so a case with one locater will be considered first. Data throughput requirements can be broken down into several categories:

- System control overhead (e.g., CAMP or other protocol)
- Physiological data
  - WPI Pulse Oximeter
  - Foster Miller Psychological Status Monitoring (PSM)
- Inertial data
- TART data

The system control overhead (the smallest portion of the total bandwidth usage) includes system commands such as requesting data captures, system function control, and debugging options. One possible method for implementing system control is to use the PPL control and messaging protocol (CAMP), which is used in the revision 2 transceiver boxes. The use of this protocol helps to maintain compatibility with currently written PPL software while still keeping a relatively small bandwidth usage. Other than large and infrequent bursts of data (e.g., sending time domain waveform configuration information), the control protocol has a conservative estimated 100bytes/sec per locator, or 0.8kbits/sec.

The physiological data is the easier for which to get an exact value. From previous work with the WPI Pulse Oximeter and Foster Miller PSM systems, near-exact data requirements are known. Both the Foster Miller PSM and the WPI Pulse Oximeter send 1sec interval update packets of a size less then 40 bytes each. The total throughput requirement for supporting both of these sensors would be 0.64kbits/sec per locator.

Experimentation using inertial sensors to improve location accuracy has been conducted recently on the PPL project. The possibility that the data associated with the inertial measurements will need to be moved over the wireless data link must be considered as a possible future expansion. The inertial data consists of a stream of samples obtained from the MEMSense nIMU[12]. A conservative estimate of throughput needs for the nIMU can be based on the packet size from the unit. The packet size is 38 bytes, and a typical stream for use in the PPL system runs at 150Hz. This would result in the need for 45.6kbits/sec for each locator.

The final portion of the data requirements is the TART data. This data requirement is by far the largest portion of throughput and the reason for the need of a new data link. The exact throughput requirements depend greatly on several variables and what preprocessing is done on the locator. The worst-case scenario will be considered first. Depending on the exact implementation of TART, there can be as few as a single transaction and as many as the number of reference antennas. Since the PPL system operates with at most 16 antennas, the maximum data that would need to be transmitted would be 16 total transactions of time domain signal data.

Equation 3.1 shows the needed throughput for a single locator sending time domain information. Equation 3.2 shows the needed throughput for a single locator sending frequency domain data. The variables for these two equations are shown in Table 3.1, along with nominal values for the equations in the PPL system. Some of the nominal values require further explanation. The number of locators in the system is a worst-case estimate both for PPL testing and for maintaining a realistic representation of a field deployable system. The location update rate has been a long-standing PPL standard at 1 second; this comes from the original system requirements defined in the PPL white paper as well as application research done with firefighters. The remaining values are dependent on the particular implementation of the PPL system. These values may change as field testing is conducted to produce the optimum balance of system performance and data throughput.

Name	**Description**	**Nominal Value**
N_{loc}	Number of locators	10
N_{rate}	Number of location updates per second	1
N_{sym}	Number of symbols to send	2
N_{trans}	Number of transactions	1 to number of reference antennas
N_{samples}	Number of time domain samples	2048 to 8192
S_{sample}	Size of single time domain sample	16bits
N_{carriers}	Number of carriers	200
S_{data}	Size of carrier data	16 to 32bits

Table 3.1: Description of TART Throughput Equation Variables

$$N_{\text{loc}} \cdot N_{\text{rate}} \cdot N_{\text{sym}} \cdot N_{\text{trans}} \cdot N_{\text{samples}} \cdot S_{\text{sample}} \tag{3.1}$$

$$N_{\text{loc}} \cdot N_{\text{rate}} \cdot N_{\text{sym}} \cdot N_{\text{trans}} \cdot N_{\text{carriers}} \cdot S_{\text{data}} \tag{3.2}$$

To develop an approximate idea of the real bandwidth needs, a case is considered with frequency domain data from one reference antenna by substituting nominal values into Equation 3.2. The result for this nominal case with 10 locators can be seen in Equation 3.3. The total system bandwidth requirements using these nominal TART values for one, three, and 10 locators can be seen in Table 3.2.

$$128,000\text{bit/sec} = 10 \cdot 1 \cdot 2 \cdot 1 \cdot 200 \cdot 32 \tag{3.3}$$

Although this gives a good idea of what a nominal case would be like, worst-case scenarios must be considered. Time domain transfers will not be considered, however, because any continuous update mode would be done with frequency domain data. Equation 3.4 shows the needed bit rate for a worst case involving sending back data for each of the 16 reference antennas in a 10-locator system.

$$2,048,000\text{bit/sec} = 10 \cdot 1 \cdot 2 \cdot 16 \cdot 200 \cdot 32 \tag{3.4}$$

By combining the above worst-case bandwidth usage, the total worst-case needed throughput for the PPL system can be found. Table 3.3 shows the total system worst-case needed throughput

Description	**1 Locator**	**3 locators**	**10 Locators**
Control Overhead	0.8kbits/sec	2.4kbits/sec	8kbits/sec
Physiological	0.64kbits/sec	1.92kbits.sec	6.4kbits/sec
Inertial	45.6kbits/sec	136.8kbits/sec	456kbits/sec
TART	12.8kbits/sec	38.4kbits/sec	128kbits/sec
Total	59.84kbits/sec	179.52kbits/sec	598.4kbits/sec

Table 3.2: Overall Data Throughput Needs (Nominal-Case TART)

Description	**1 Locator**	**3 locators**	**10 Locators**
Control Overhead	0.8kbits/sec	2.4kbits/sec	8kbits/sec
Physiological	0.64kbits/sec	1.92kbits/sec	6.4kbits/sec
Inertial	45.6kbits/sec	136.8kbits/sec	456kbits/sec
TART	204.8kbits/sec	614.4kbits/sec	2,048kbits/sec
Total	251.84kbits/sec	755.52kbits/sec	2,518.4kbits/sec

Table 3.3: Overall Data Throughput Needs (Worst-Case TART)

for one, three, and 10 locators.

From Table 3.3, it can be seen that about 2.5Mbits/sec of throughput would be needed to support 10 locators with physiological and internal data. Although this number is relativity high, it is unlikely that during the course of testing with the PPL system, 10 locators with full physiological monitoring in conjunction with inertial data will be used.

It is also important to note that this throughput is associated only with the PPL-specific data. In addition to this throughput, considerations must be made for error correction data, communications protocol overhead, and retransmission for guaranteed arrival.

3.2.2 Range Requirements of High speed Data Link

In the testing of the PPL system, the digital data link must provide reliable coverage over the entire test location. Because the PPL project is a dynamic endeavor, not all situations it will encounter can be considered. However, some requirements can be drawn from past experience.

First, the data link should not be the limiting bottleneck for the system. If the RF ranging signal is able to penetrate and be usable in a location, the data link also must be reliable in this location. PPL testing has taken place in various types of buildings, including many around the WPI campus. These locations range from multi-floor academic buildings on the WPI campus to three-story wooden residential buildings. In addition to buildings, there has been testing at large-scale outdoor line-of-sight locations up to 1km square in size.

In the outdoor locations, the data link must provide coverage over distances of up to 1km. For indoor locations, the data link must be able to cover medium-size, multi-floor, commercial buildings. For instance, testing has been conduced on the WPI campus at various buildings, including individual wings of Atwater-Kent, the Bartlett Center, and the campus ministry center. Other locations where testing has been conducted are various firefighter training locations. Although some of the firefighter training buildings are constructed entirely of metal and consequently impenetrable by either the data link or the RF ranging signal, others are constructed of thick concrete walls with heat-resistant tiling on the interior. The possible use of the data link in these locations also must be considered in the selection process.

3.2.3 Data Link Solution

One possible solution considered for the high speed data link was the ZICM2410P2 ZigBee–based wireless module produced by California Eastern Laboratories (CEL)[13]. The ZigBee module provides several features that are appealing for thhi application. The module is a 2.4GHz chip with a maximum power output of 20dBm and maximum data bandwidth of 1Mbit/s.

The use of the 2.4GHz band when compared with the Xemics module's 915MHz band has some obvious detractions. The 2.4GHz will have trouble penetrating RF obstructions as deep as a 915MHz signal. However, the CEL chip has other features that help make up for this inadequacy. The Xemics module has a maximum power output of 15dBm. The higher power of the CEL module, 20dBm, will aid in compensating for the higher frequency band of operation. One other advantage is the use of the ZigBee protocol. The included ZigBee software network stack allows for quick implementation of robust mesh networks. Through the use of mesh networking and overlapping fail-over communications, the effective coverage of the CEL chip can be extended by simply placing extra receiver units around a test location.

When the ZICM2410 is used, an interesting method can increase the effective usable bandwidth. Although the maximum bit rate for the ZICM2410 is 1Mbit/s, this can be extended through a method referred to as channel bonding. Up to four frequency orthogonal channels are available with the ZICM2410 that can be used concurrently without harmful interference. By placing various mobile locators on separate channels, a boost of 4x total bandwidth can be obtained. This extra bandwidth will allow for more concurrent devices to be used or for a reduced bandwidth operation for higher link robustness.

Although the ZICM2410 provides the bandwidth needed for the PPL system, it sacrifices link robustness and signal penetration due to the higher bandwidth and higher frequency band. For that reason, it was decided to include the Xemics XE1205–based module in the system alongside the CEL module. The CEL module in preliminary testing with development boards showed significantly less penetration through buildings compared with the Xemics module. By using the Xemics module as a command channel backup, an active link can be maintained with mobile locators even when the CEL module has lost connection. Additionally, the software integration for the Xemics module had been completed on previous revisions of the locator, which will allow for a low overhead during software system development.

3.3 Transmit and Receive Channels

The transmit and receive channels are two of the most fundamentally important systems in the revision 3 hardware design. Previous revisions of both locator and transceiver hardware provide references for ADC and DAC selection. The pros and cons of these implementations helped greatly when the design for the new locator was being considered. Additionally, the analog chains, including the amplifiers and filters, have been revisited to meet the new requirements. Before the selection of the DAC device and design of the related analog circuitry, the requirements for the transmitted waveform must be clearly defined.

3.3.1 Transmitted Waveform

One of the key pieces to the PPL system is the transmission of a multicarrier waveform. In previous designs, there have been several overlooked or under performing parts of the transmission system. The goal in the redesign process was to identify fully all the requirements of the transmission system and achieve the best possible hardware realization.

The ideal waveform used by the PPL locator can be defined as a periodic multicarrier signal that is composed of a number of equally spaced sinusoids in the frequency domain. The frequency range of the signal is defined by the FCC license that was filed for by the PPL project. The PPL system gains performance increases with more bandwidth, so a block of frequencies with large bandwidth was selected. The allotted frequencies for the experimental license granted by the FCC run from 512–698MHz, with a restricted area from 608–614MHz. The 512–698MHz region was selected both for the amount of bandwidth available and for the location of the band in the spectrum. The restricted area of operation, 608–614MHz, was a section of the spectrum reserved for emergency communications. An example waveform regenerated by the revision 2 hardware and captured on a spectrum analyzer can be seen in Figure 3.4. This waveform ranges from 600–690MHz except in the 608–614MHz region, where transmission is restricted.

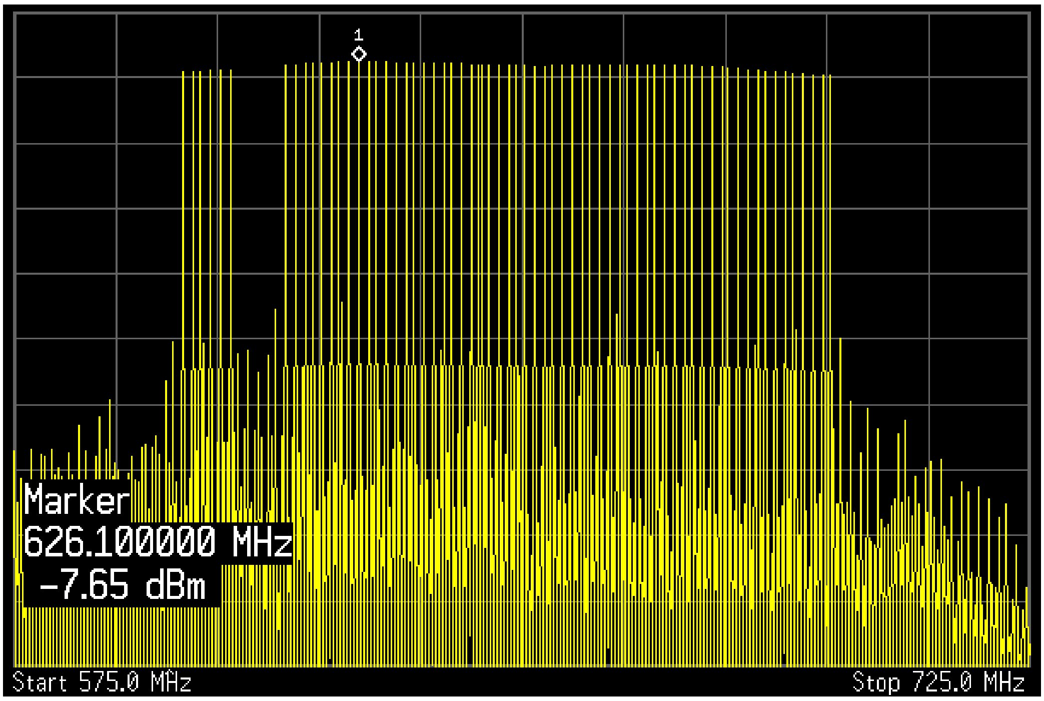


Figure 3.4: RF Multicarrier Waveform – Revision 2 PPL Locator Hardware

FCC license

The redesign needs to take into account the FCC license that is in effect for this project. Currently, the license allows for operation from 512–608MHz and 614–698MHz. In each of those two bands, the maximum effective radiated power (ERP) is 800mW. The use of ERP in the license means that antenna gain and efficiency must be accounted for in the design. Although the current license is a special temporary authorization (STA), in the near future application will be made for a permanent test license, which likely will have the same restrictions and allowances.

One other consideration in regard to legality and the FCC is the allowed out-of-band emissions. Out-of-band emissions are not governed by our specific license but rather by a set of laws that cover out-of-band emissions in the PPL signal's frequency range. This FCC law states that an ERP of –49dBm per 200kHz span out-of-band emission must not be exceeded. If the standard PPL reference antennas are used, which have approximately a 3dBi antenna gain, the power to the antenna would need to be 3dB lower to have a legal ERP. This out-of-band emission allowance applies to the areas both below 512MHz and above 698MHz, as well as the restricted band of 608–614MHz.

Previous Issues With Transmitted Signal

The shortcomings of the transmission systems of previous generations of hardware can be grouped into two major categories: illegal out-of-band emissions and limited total transmitted power.

The first issue, out-of-band emissions, has been observed in previous revisions of hardware in two ways. First, there is the issue of IP3 distortion. IP3 distortion is a result of amplifying the multicarrier signal and is interleaved inside our transmission band as well as on the edges of our band. This is a particular issue in the restricted band from 608–614MHz inside our allotted frequency range. An example of IP3 distortion in this area from a previous generation of hardware can be seen in Figure 3.5. The lower amplitude peaks in the middle of the capture are a direct result of IP3 distortion.

The second out-of-band emissions issue that has presented in the past is leakage of the mixer local oscillator (LO) to the RF output. The RF mixer system was designed to use an LO of 730MHz, but that resulted in excessive leakage, which was the result of two factors. First, the LO is relatively

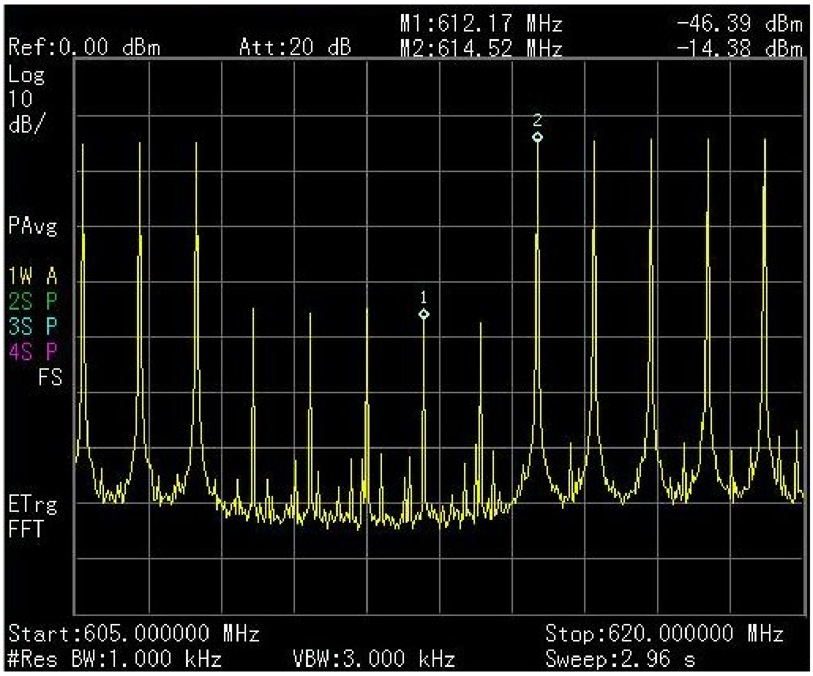


Figure 3.5: IP3 Distortion in the Restricted Band

high power compared to the individual carrier strength. Individual carrier strength from baseband can be as low as –30dBm before amplification and filtering, depending on the carrier count, the density, and the crest factor. The LO is approximately –2dBm before amplification and filtering. The intent was that the LO would be attenuated significantly by the band pass filters (BPF) of the RF amplification stage. However, it was found that there was excessive leakage of the LO past the legal limit to the antenna, an effect that can be seen in Figure 3.6. The characteristics of BPF roll-off can be seen in Figure 3.7.

These issues relating to FCC non compliance were solved with rather simple solutions. Lowering the output power of the transmission by reducing usage of the DAC's dynamic range alleviated each of these issues. As the generated signal is lowered a set value, the IP3 distortion will drop three times this value. For example, a drop of 3dB in the generated signal will result in the IP3 distortion dropping by 9dB. This effect allows for a marginal loss in total signal power to drop the IP3 distortion below the FCC requirements.

The second issue of the LO leakage was solved by moving the LO mixer carrier further up in the

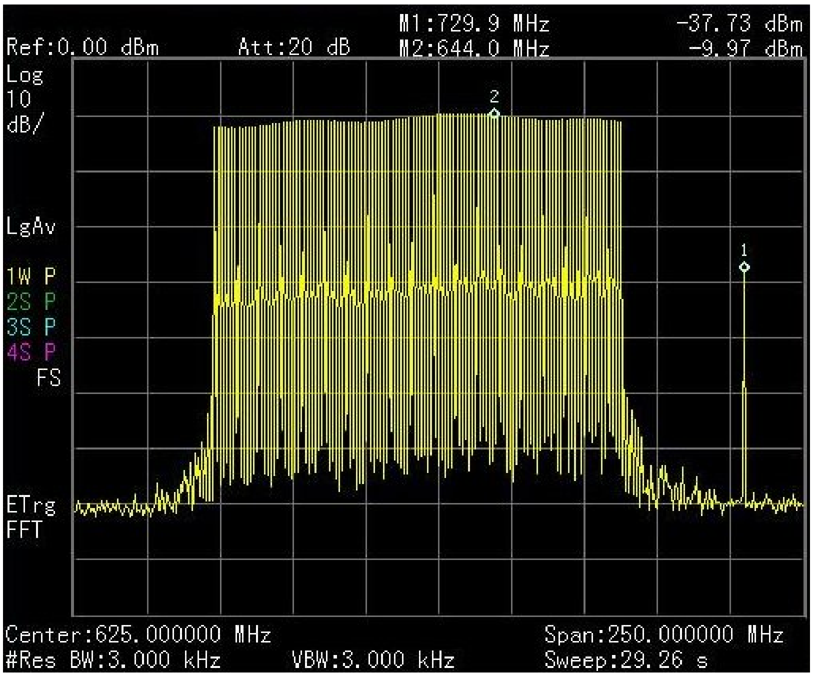


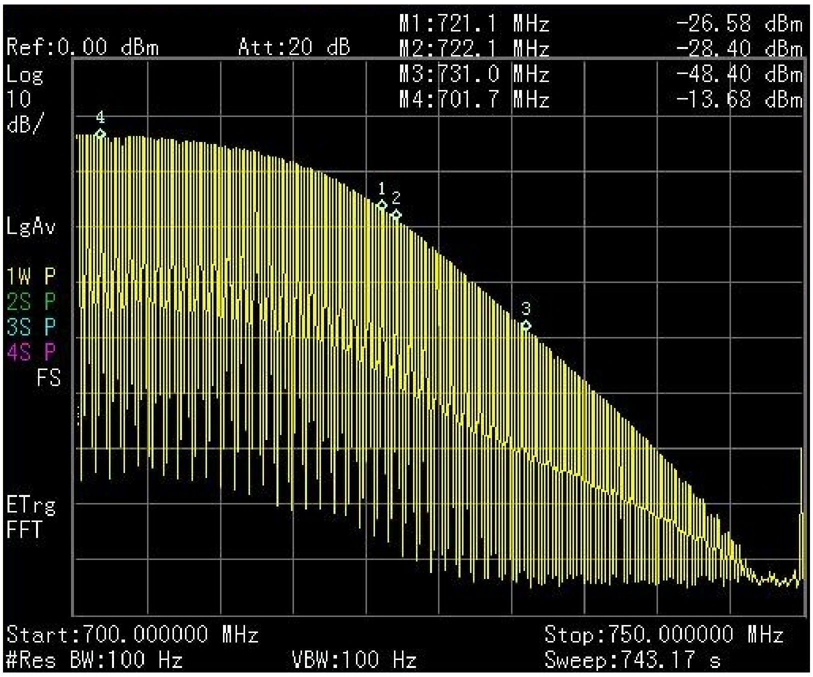
Figure 3.6: 730MHz LO Leakage at RF Output

spectrum to 750MHz. At 750MHz, the attenuation provided by the band pass filters was adequate to suppress the LO carrier below the FCC requirements. This method of compliance was not without its drawbacks to the system. By moving the LO carrier to 750MHz, imaging artifacts from the DAC output moved up in frequency enough such that they no longer were suppressed sufficiently by the band pass filter. To combat this effect, the total usable bandwidth of the PPL multicarrier signal was reduced to remove these imaging artifacts. Approximately 15MHz of bandwidth was lost to this fix.

Although these issues did require some sacrifices to achieve FCC compliance, the revision 2 hardware still was capable of successful operation. However, it was important that these issues were addressed during the redesign of the revision 3 hardware.

Final Waveform Requirements

From the above issues and limitations, a set of requirements was generated for the signal synthesis and RF amplification hardware. The issues mentioned above all needed to be addressed to produce a successful transmission block for the new locator.



Ref:0.00 dBm Att:20 dB

M1:721.1 MHz -26.58 dBm
M2:722.1 MHz -28.40 dBm
M3:731.0 MHz -48.40 dBm
M4:701.7 MHz -13.68 dBm

Start:700.000000 MHz Stop:750.000000 MHz
#Res BW:100 Hz VBW:100 Hz Sweep:743.17 s

Figure 3.7: Roll-Off Characteristics of Band Pass Filter

The list of requirements is as follows:

- Low in-band IP3 distortion
- Necessary out-of-band image filtering to comply with FCC regulations
- Suppression of LO leakage to comply with FCC out-of-band emissions
- At least the same power levels for transmission as previous revision

3.3.2 Digital to Analog Converter

From the list of requirements for the transmitted signal shown above, a Digital to Analog Converter (DAC) that satisfies these must be selected. In all previous mobile locators, an Analog Devices AD9726 DAC was used. During component selection, it was found that newer DACs were available that better suit the needs of this application.

For this revision, an interpolating I/Q DAC, an AD9122, was selected. Although the implementation and use of an I/Q DAC is more complex compared to a regular DAC, several of the major issues in the transmit chain can be addressed with its use. The inner workings of the DAC are documented in Section 4.4, but a brief overview of its advantages are covered here.

The interpolating nature of the DAC removes the issues relating to the DAC image filtering. The interpolation moves the higher images double the distance away allowing for easier filtering. Additionally, the LO leakage is addressed though use of auxiliary DACs that can be used to suppress the LO spike with active feedback.

3.3.3 Analog to Digital Converter

The use of a receive channel on the mobile locator hardware to support TART was a first for the PPL project in this design. From previous work with receive channels in the reference transceiver hardware, a base ADC design can be found. The current revision of PPL reference transceivers has four independent receive channels with a Texas Instruments ADS5474 ADC at the core of each. The ADS5474 is a high-speed, 400MSPS, 14-bit ADC[14]. This design was found to meet the needs of the PPL project throughout its use. The design was also used during preliminary proof-of-concept testing of the TART algorithm. With a proven design already implemented, the reuse of the previous receiver system block for the new locator was chosen as the best option.

3.3.4 Filtering and Analog Chains

The analog chains of both the ADC and the DAC have seen changes from the previous revision as well. In the time since revision 2, newer analog amplification components have become available that better suited the needs of the locator device. Additionally, the filtering methods used to condition the transmitted and received RF signals required modification. In previous revisions of the mobile locator, as well as in the PPL transceiver, additional band pass filters were reworked into the design to achieve compliance with FCC requirements. These reworks have been included in the revision 3 hardware.

The reworks center around the addition of an extra band pass filter. In the revision 2 hardware, only one band pass filter was used in the receive chain and one in the transmit chain. The single band pass filter provided insufficient suppression of out-of-band signal.

To combat this issue, an additional band pass filter was added to the design. Both the transmit and the receive chains retained their band pass filters; however, an additional filter was placed directly before the antenna to provide additional out-of-band suppression (Figure 3.8).

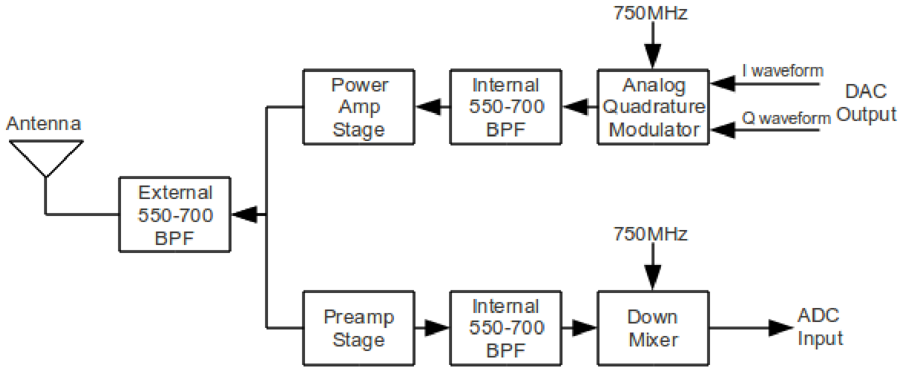


Figure 3.8: Updated Filtering Scheme – Revision 3

3.4 FPGA

At the heart of the TART locator is a Xilinx FPGA. In revision 3, the FPGA is used to communicate over the data link, coordinate and move high-speed transmitted and received digital data, perform preprocessing of data, and carry out various other tasks known and unknown as of yet.

In revision 2, two different FPGAs were used. The first FPGA located on the DWG PCB was a small high speed grade Spartan 3 device used exclusively for waveform generation. The FPGA architecture performed two basic tasks. First, it managed the startup sequence of the DAC and associated components such as the PLLs. Second, once initialized, it streamed digital samples to the DAC to produce the baseband waveform. The second FPGA was a newer Spartan 3A series FPGA and was responsible for overall system control and digital communication on the data channel.

In the revision 3 hardware, PCB space and system power are at a premium because of the addition of new system functions and many components, such as the receive channel. To address the issues of board space and power consumption, the possibility of condensing the functionality of the two discrete FPGAs into one unit was considered as a possible solution.

Since the design and implementation of the revision 2 hardware, Xilinx introduced a new line of Spartan FPGAs, the Spartan 6. This previously unavailable technology was considered as an option for inclusion in the new system. Blindly choosing new technology simply because it has become available, however, is not a proper way to approach the design process.

Before considering the Spartan 6, the reasons for the two FPGA-based approach in revision 2 need to be reviewed. In revision 1, a single high speed, low gate count Spartan 3 device was used to drive the DAC for signal generation on the DWG. In the revision 2 design process, it was decided that the digital system should also be FPGA based. During that design process, it was much simpler to keep the existing Spartan 3 design for signal generation since the DWG and the data channel were kept highly segregated. The newer Spartan 3A was used on the data channel to implement the digital control system (Figure 3.9).

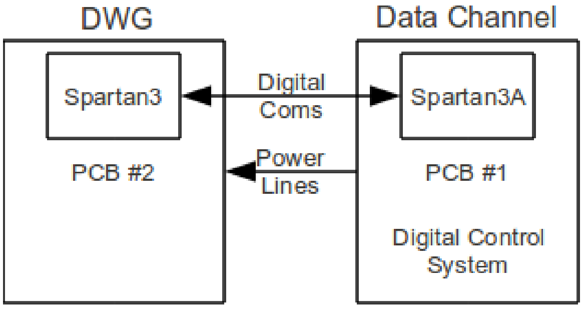


Figure 3.9: Revision 2 Boards Showing the FPGA Layout and Interfacing

In the design process of the revision 3 hardware, the high level of segregation that existed between the data channel and the DWG no longer fit the needs of the PPL project due to the implementation of a full transceiver design. With the inclusion of a receive channel as well as an I/Q DAC, the RF hardware requires direct interfacing with the real-time FPGA system in order to operate. Two segregated FPGA systems could no longer provide the necessary level of control that is required with the new system architecture.

With this distinct design departure from previous revisions, revision 3 condensed all FPGA operations to a single device to address these system requirements. This design decision resulted in lower power consumption, fewer components for a smaller PCB, simpler FPGA design and coding, an overall simpler, easier-to-maintain system as well as directly addressing the needs of the new system architecture.

With the decision made for a single-FPGA approach, the selection of which FPGA to use was considered. For all previous FPGA needs of the PPL project, Xilinx has been the de facto choice. The FPGA offerings from other companies compare very closely with Xilinx's lineup of chips. To avoid an unnecessary learning curve, Xilinx remained the choice of FPGA for the PPL project.

The decision of which Xilinx FPGA to use comes down to the question of whether the design costs of introducing a new series of chip are outweighed by the gains it can offer. The Spartan 6 series offers many advantages over the Spartan 3 and Spartan 3A chips used in previous designs. As has been the trend with newly introduced FPGA devices, there are several areas that are incrementally improved from series to series. Among these are lowered power consumption, smaller chip footprint, increased on-chip resources (RAM, logic gates, etc.), and a new product life cycle for longer support from manufacture into the future.

A comparison of the top-end Spartan 3A device with the top-end Spartan 6 device is shown in Table 3.4. As can be seen in the table, the specifications for the Spartan 6 exceed the Spartan 3A's specifications. Key to the PPL project is the inclusion of almost double the block RAM and the high count of digital signal processing (DSP) blocks. The large amounts of block RAM is used in the PPL project for digital waveform storage as well as in the Microblaze system for code and data memory. The DSP blocks are used in the implementation of real-time signal processing in hardware. For the PPL project, large quantiles of both these resources are desired due to easier implementation of FPGA design and anticipation of future system expansion.

	Spartan 3A (XC3SD3400A)	**Spartan 6(XC6SLX150)**
Logic Cells	53,712	147,443
Block RAM (Kb)	2,268	4,824
Digital Clock Managers	8	12
DSP Blocks	126	180
Maximum Block RAM Operating Frequency (MHz)	280[15]	260[16]

Table 3.4: Spartan 3A vs Spartan 6

Spartan 3A vs Spartan 6[5][6]

The advantages of the Spartan 6 are clear; however, the design costs must be considered. The Spartan 6 is a direct continuation of the Spartan line of FPGAs, and implementation is nearly identical to that of a Spartan 3 or Spartan 3A system. Along with the similarity in design, Xilinx provides reference designs to speed implementation of Spartan-based systems.

With the Spartan 6 chosen as the FPGA for the revision 3 hardware, all that was left was to select the chip package. For this design, the LX100 and LX150 were chosen, mainly governed by the fact that these two parts offer the maximum block RAM of the Spartan 6 family. These two parts are available in four different chip packaging options, as shown in Table 3.5.

Package	Size (mm)	Ball Spacing (mm)
CSG484	19 x 19	0.8
FG(G)484	23 x 23	1.0
FG(G)676	27 x 27	1.0
FG(G)900	31 x 31	1.0

Table 3.5: Spartan 6 LX100 and LX150 Package Options[6]

The final Spartan 6 chosen was the Spartan 6 XC6SLX[100/150] in CSG484 packaging (chosen for its small size). The decision to move to a single FPGA has advantages in both system complexity and PCB usage. The Spartan 6 proved to be the best option for this single FPGA implementation and the package was chosen to reduce the PCB footprint. This design decision results in a revision 3 hardware that will be based around a single Spartan 6 FPGA at the core of the digital system. Figure 3.10 shows how the previous revisions' segregated systems fit together in the new revision 3 hardware.

3.5 Power and Battery

In the revision 2 locator, a 7.4 volt 4.8 amp hour (35.5 watt hour) lithium-ion battery pack was used as the mobile power source. A battery-charging system was included on the PCB that allowed for charging of the battery while the system is in use. In addition, the system can be used without a battery, running directly off a DC power source. Throughout the use of the revision 2 hardware, this battery proved it could provide the needed power to keep the locator hardware running during all field testing.

With the addition of a receive chain in the revision 3 hardware, there is a need for a higher capacity battery power source. To cope with this need, a new lithium-ion battery pack was selected

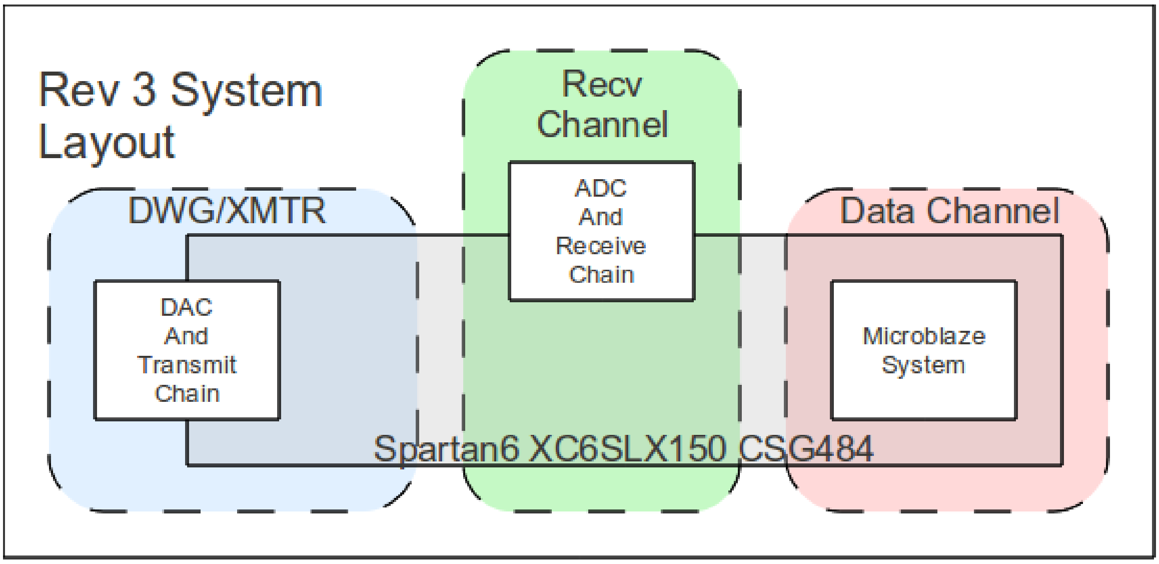


Figure 3.10: Revision 3 FPGA System Layout

for use in the revision 3 hardware. A 11.1 volt 5.2 amp hour (57.8 watt hour) battery pack was selected for its higher power capacity compared to the 35 watt hour battery used in revision 2. This new unit has approximately 66% more power capacity and physical size of the previous battery pack.

Extensive power consumption research was conducted on the revision 2 locator hardware. The revision 3 hardware shares many similar components and designs elements with the revision 2 hardware. An estimate of the revision 3 hardware battery consumption can be extrapolated from this previous testing while accounting for new hardware such as the ADC. System battery usage is heavily dependent on the duty cycling of high power consumption components in the final software design. Although these duty cycle values were not known during this design phase, they can be conservatively estimated based on previous experience. The revision 2 hardware was capable of operating with a 10% transmit duty cycle for time spans of well over eight hours. The addition of the ADC channel will result in power consumption that can be compared to that of the transmit channel. Additionally, a 10% duty cycle is a fair conservative estimate for the ADC and receive channel. With the addition of almost double the power capacity and accounting for the power savings of a single FPGA design, it is concluded that the selected battery will be able to operate a mobile locator for time spans in excess of eight hours, well beyond the needs of PPL field testing.

The charging interface that was included in revision 2 has been modified to work with the new higher capacity battery. The unit will have a DC power plug that will take an 18 volt power source that can provide a minimum of 2 amps. The unit can run directly off this power source while simultaneously charging the battery pack and can be hot plugged while the unit is running, much like the previous revision.

3.6 High Speed Memory

In the new locator, there is a need to store the information of the captured waveform. A high speed memory device is needed that can keep pace with the samples coming from the ADC. In the revision 2 PPL transceiver, several high speed SRAM chips were used to satisfy this need.

Although, SRAM is a distinct possibility for inclusion in the revision 3 hardware, other options were considered because SRAM is an expensive and high power consumption device. One possible alternative technology to SRAM is DDR SDRAM, which is the same type of memory used in personal computers for general system RAM. However, due to the refresh cycles and complicated interfacing associated with DDR SDRAM, it was decided that SDRAM would not be a suitable technology for a real-time, speed-critical application such as PPL.

A new SRAM chip, the Cypress CY7C1380D, was selected for use in the revision 3 hardware. This SRAM chip is arranged in a 512k x 36 configuration, providing 18Mbit of total storage. However, because this device is used exclusively for the storage of ADC data samples, not all 36 data lines are connected. Each ADC sample is 14bits wide. Since the SRAM provides 36-bit wide data bus, two samples can be stored at each memory address. To reduce total I/O lines on the FPGA, only 28 data lines are connected, allowing for two ADC samples to be stored at each address. The remaining unused data lines are tied to ground (Figure 3.11). A total of 1024 total ADC samples can be stored in SRAM at 100% capacity. In addition, storing two samples at each memory address allows for real-time access speeds to the SRAM to be at half the incoming data rate of the ADC. To handle the ADC's data rate of 440MSPS, the SRAM will need to operate at 220MHz with a two-sample-wide data bus. A 250MHz version of the SRAM was selected to meet this timing requirement.

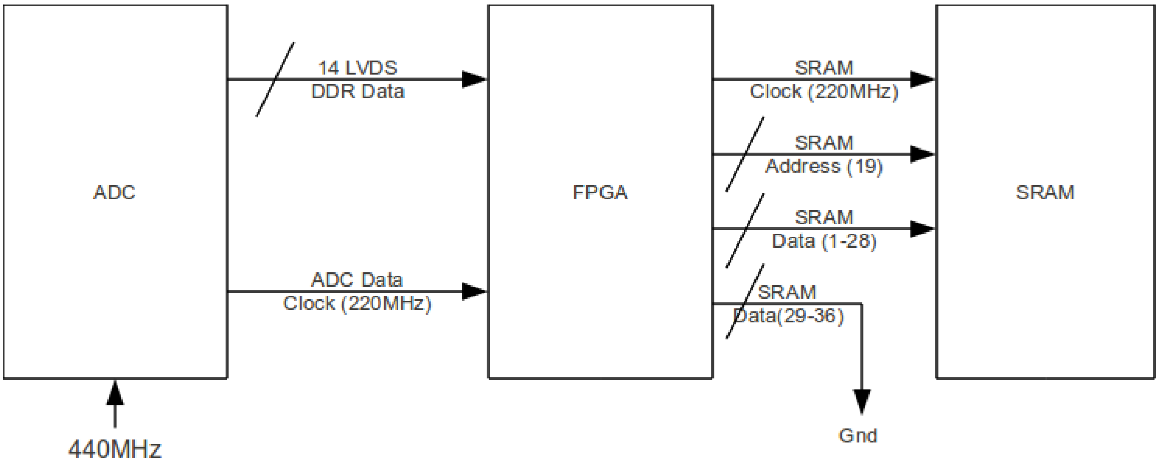


Figure 3.11: ADC and FPGA, SRAM Interface

3.7 Thermal and Size Considerations

Throughout the entire design of the new locator, both the size and the thermal properties of the device need to be considered. Because this device will be used as a mobile locator as well as a reference unit, the total size of the PCB must be kept on a manageable scale for mounting on personnel. Each component selection was weighed heavily for its size impact to the total PCB.

The mobile device will be mounted in as compact an enclosure as possible. For this reason, the thermal properties of certain sections of the design must be considered. The two sections of the design that produce the most heat are the transmit and receive chains. Although the transmit chain has the highest heat dissipation, it will be duty cycled at such a small percentage that the total heat produced will not exceed the devices' thermal tolerance. The ADC in the receive chain also produces a large amount of thermal output. In the mobile device, this section will also be duty cycled at a small percentage, rendering heat production a non-issue. To help with heat dissipation, the PCB has been designed to accept heat sinks at these thermal critical locations.

For units that are used as reference, size is much less of a consideration. Although the PCB hardware will be identical to that used in the mobile locators, a more forgiving enclosure can be used. Both the transmit and the receive chains are required to operate at higher duty cycles in the reference units compared with the mobile locators. However, with the relaxed enclosure size requirements, this obstacle can be overcome with the inclusion of larger heat sinks and cooling fans.

3.8 Peripheral Components

In addition to the major functional systems of the locator, several smaller discrete components are included in the design. Many of these components have been included in previous revisions of the locator hardware. Various components have been added or replaced with functionally superior equivalents.

3.8.1 nIMU Interface

Since 2008, the PPL project team has experimented with the integration of high precision inertial sensors for location estimation. The first sensor used for this purpose was a MEMSense nIMU (Figure 3.12). Initially, testing with this unit was performed in the lab with temporary interfaces that were constructed solely for this purpose[17].

Figure 3.12: nIMU Inertial Unit

During the design phase of the revision 2 hardware, the nIMU had reached a high level of usage in PPL research. A dedicated interface for the nIMU was included in the design to ease the integration of this sensor with the system. The need for this interface is due to the nIMU's use of a voltage range not used anywhere else in the locator system. In addition, an I^{2}C interface is used for communications that requires physical pull-up resistors. The interface in revision 3 is functionally identical to the interface in revision 2 (Figure 3.13).

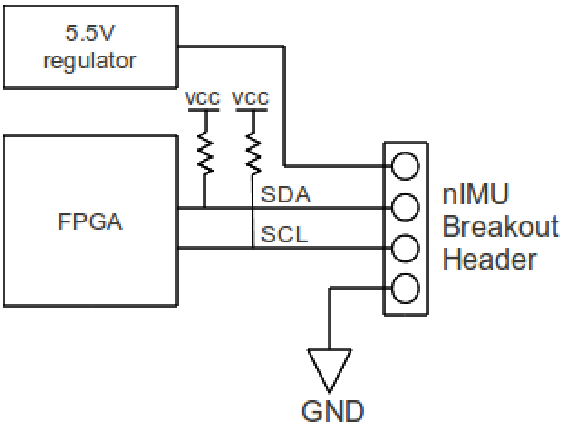


Figure 3.13: nIMU Sensor I^{2}C & Power Interface

3.8.2 NavChip Inertial Unit

In continuing with the inertial research on the PPL project, various other sensors have been tested. One of these sensors is the InterSense NavChip. This device offers functionality similar to that of the MemSense nIMU. During the design of the revision 3 hardware, there was a lack of sufficient testing with this new device to determine its final inclusion in the system. The decision was made to include a footprint for the PCB-mounted sensor on the new hardware.[18].

3.8.3 Barometric Pressure Sensor

Barometric pressure sensors have been used in the past to help provide more accurate estimations of vertical location. This information is used to provide an estimate of the particular floor of a building on which a first responder is located. Knowing which floor the first responder is located on is crucial information.

In the revision 2 hardware, an analog pressure transducer was paired with a 24-bit ADC for pressure capture. There were several issues that came with this configuration. First, the pressure transducer suffered from a major uncompensated temperature drift. This drift was most pronounced when a board was first powered on and the ambient temperature of the PCB and enclosure slowly raised. This necessitated a waiting period of nearly 20 minutes before the pressure values stabilized and were usable. Second, the sensor was found to not provide data accurate enough to help estimate vertical position at the precision the PPL project requires.

In the new design, a stand-alone chip, the Bosch BMP085, with a digital interface was chosen to replace the previous pressure sensor. This new sensor provides several features that address the above-mentioned issues. The unit contains an on-chip temperature sensor that is used in the calculation of the pressure value, which eliminates any drift related to temperature and allows the pressure values to be used reliably from the moment of power on. Additionally, the new sensor boasts more accurate and stable pressure readings compared to those of the previous revision configuration.

3.8.4 MicroSD Memory Card Interface

During lab testing with previous revisions of mobile locator hardware, one area that has proved problematic is storage of large amounts of sensor data. Several situations require the storing of large amounts of data, such as long-term barometric data captures and inertial data captures.

Revision 2 included a 128Mb SPI EEPROM that was used for storage of the FPGA configuration data. However, the configuration data occupied only the lowest 2Mb of the EEPROM. The remaining 126Mb was intended for the purpose of nonvolatile data storage for situations such as the above-mentioned data captures. However, the EEPROM proved to be an inadequate solution due to slow interfacing, the need for page erasure, and complications relating to retrieving the data to computer workstations for analysis.

A MicroSD card interface was included in the revision 3 design to address this issue. The MicroSD card has a small PCB footprint while providing nearly limitless data storage. The MicroSD cards in this system are interfaced though MMC protocol over an SPI-like interface. From a software standpoint, this allows for easy integration with the system. In addition, the interface can operate at speeds of up to 25MHz, providing write speeds that are orders of magnitude faster then the EEPROM. The final convenience provided by the MicroSD card interface is the ease with which data is retrieved. The card can simply be removed from the locator and downloaded directly to a computer though a MicroSD card reader for analysis.

3.8.5 PPL Small Radio Module

As mentioned in Section 1.2, the PPL project has explored integration of personal physiological sensors into the mobile locator hardware. Two such systems have been integrated into the PPL system thus far: the WPI Pulse Oximeter oxygen saturation monitor and the Foster Miller Physiological Status Monitoring (PSM).

Both devices are designed to communicate in a personnel area network (PAN) based on a single radio receiver integrated with the mobile locator. These devices all operate with a Chipcon CC1101, 915MHz, 10dBm radio module. Each device provides low frequency updates (approximately once a second) of the physiological data. The data is then relayed to the base station through the high speed data link.

The PPL small radio module (SRM) interfaces with the FPGA through a standard UART interface along with various chip-specific control lines. A footprint for the SRM module was included in the revision 3 hardware design. However, only hardware intended for use as a mobile locator, as opposed to reference usage, will require the SRM to be populated on the board.

3.8.6 Ethernet Interface

With the new locator hardware being used for both mobile and reference applications, there is a need for a reliable, wired, high speed data interface. In the previous implementation of the system, the rack mount reference transceivers each had a gigabit Ethernet interface for communications purposes. The usefulness of this Ethernet interface for communication during field testing as well as debugging and development in the lab was invaluable. For those reasons, it was decided to include an Ethernet interface on the new locator.

In the transceiver boxes, a gigabit Ethernet PHY was used with the media access control (MAC) implemented on the Vertex-4 FPGA. Although this chip served its purpose in the transceiver design, it is physically too large to include on a mobile platform. Additionally, for the purposes of the PPL project, gigabit communications are not required. As a possible way of reducing the PCB footprint of the Ethernet chip, 100 megabit MAC/PHY combinations were considered.

A 100 megabit Ethernet chip was selected for use on the mobile locator. This chip, the Micrel KSZ8851, provides several advantages over the gigabit implementation in the PPL transceivers.

First, physically the chip is significantly smaller and is appropriate for inclusion in this mobile design. Second, the chip includes a full MAC as well as a PHY implemented in hardware. With a built-in MAC on the chip, there is no need to include a MAC implementation on the FPGA, which has led to problems in the past with the PPL transceivers. Rather than providing a standard Ethernet PHY interface, this chip provides a generic communications interface with control lines and a parallel data bus that can be dynamically switched to an 8-bit or 16-bit width.

The Ethernet interface has several intended uses for the new revision hardware. During initial system implementation and laboratory debugging/testing, the interface will be used to move data, such as captured waveforms, at high speeds to and from the device. In a field testing configuration, the interface can be used as a reliable fallback from the CEL wireless interface for reference unit communication to the base station. The Ethernet interface allows for either direct CAT-5 wiring to the base station or connection to a WiFi-based communications system much like what was used with the PPL transceivers.

3.8.7 Low Precision Accelerometer

It was decided that a low precision, 3-axis accelerometer be included in the system. The accelerometer chip is the same Analog Devices ADXL330 that was included in revisions 1 and 2, in tandem with an ADC for digital interface to the FPGA. It outputs independent X/Y/Z acceleration information with a ±3g range[19].

A slight change has been made to the implementation of this chip compared with the previous revisions. In previous revisions, the three analog axis channels were summed and integrated before delivery to the ADC. This processing provided a metric of total movement in any direction. The metric was then used to ascertain if the unit had remained stationary for a period of time and flagged a warning signal much like a Personal Alert Safety System (PASS)[20].

The new revision induces a multichannel ADC that will have each of the three channels connected independently. This allows for the microcontroller to analyze the raw data from all three channels rather then relying on a fixed processing implemented in hardware. The data will still be able to be used in the same manner as in previous revisions for the purpose of emulating PASS-type functionality. In addition, the raw data can be used from reference stations to determine the orientation of the device, which can be used to extrapolate antenna placement.

3.8.8 Battery Voltage Sensor

In all previous revisions of mobile locator hardware, an interface for sensing the system's battery voltage was included to alert the base station of imminent battery depletion. The battery source voltage is too high for an ADC running from system power to read. For that reason, a simple voltage divider is used to condition the battery voltage into a range appropriate for an ADC to sample. The multichannel ADC used for sampling the low precision accelerometer had remaining unused channels, one of which is used to sample the conditioned battery voltage (Figure 3.14.

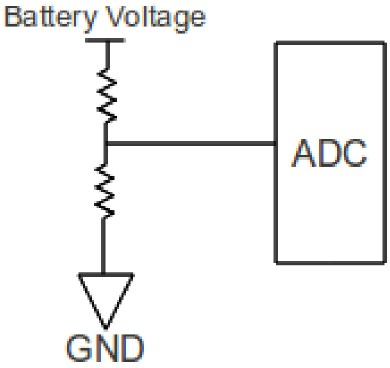


Figure 3.14: Battery Voltage Sensor

3.8.9 Physical Unique ID Switch

Included in this design is a bank of eight DIP switches. Each switch is independently wired to the FPGA, providing a parallel 8-bit value to the system. These switches are included for several reasons.

First, the switches will be used in a final testing system to identify each piece of locator hardware with a unique ID number. From this unique ID number, the function of the unit (e.g., mobile locator or reference unit) can be determined so the proper set of microcontroller code can be executed. Also, system settings such as the Ethernet MAC address, the IP address, and other system identifiers can be determined by the DIP switch settings.

Second, the switches can be used to allow for field-configurable options, including transmission power, transmitted signal selection, and various other modes of operation.

Finally, these switches will be used during the development stage of HDL and software, providing an invaluable interface for debugging and bench testing prior to deployment in the field.

3.8.10 GPIO Header and LEDs

As with any complex system, considerations must be made in this new revision for debugging and future expansion. Two of the most valuable tools for debugging in a new system are general purpose input/outputs (GPIO) and LEDs. Both of these tools can be used during the implementation and debugging process to help convey information about the systems status to the outside world.

A total of eight GPIO lines are included in a 2 by 8 header. Each of the eight GPIO lines is complemented with a ground pin. In addition to use for debugging, the GPIO header is included to provide an interface for connecting future peripherals. The mobile locator hardware has shown itself to be an appropriate platform for testing new peripherals before inclusion in a future revision.

Two board-mounted LEDs are included solely for debugging purposes since they will be obscured when the PCB stack is mounted within a case. In addition to the board-mounted LEDs, interfaces for two bidirectional LEDs are included for the mounting of two bi-color LEDs. The LEDs, whose purpose is to convey system status, are intended to be visible on the outside of the enclosure case.

Chapter 4

System Design and Implementation

The previous chapter explained the system level design as well as various component selections for the subsystems. With the system level design and component selection completed, the next step was to plan the theoretical operation and implementation of the individual subsystems. In addition to the physical hardware systems described in Chapter 3, the digital systems implemented on the FPGA, including the Microblaze soft-core processor and high-speed data interfaces, needed to be thought out.

A component-level block diagram of the implementation representing the design decisions laid out in the previous chapter is shown in Figure 4.1. Key among the hardware systems is the transceiver consisting of the transmit and receive chains. With an entirely new I/Q DAC used in this design, theoretical operation, digital waveform creation, and device interfacing all needed to be designed from the ground up. The receive channel, with its revision 2 transceiver-based design, required less novel work to implement. The operation of the digital data link, specifically the Xemics 915MHz module used for system control, as well as the protocol used for its operation needed to be designed as well.

The following sections describe the implementation and operation of the major hardware subsystems. The first subsystem to consider is the FPGA, which is at the center of the revision 3 hardware. The FPGA digital system consists of the high-speed interfaces to the DAC and ADC, hardware implemented processing blocks, as well as the soft-core Microblaze processor system.

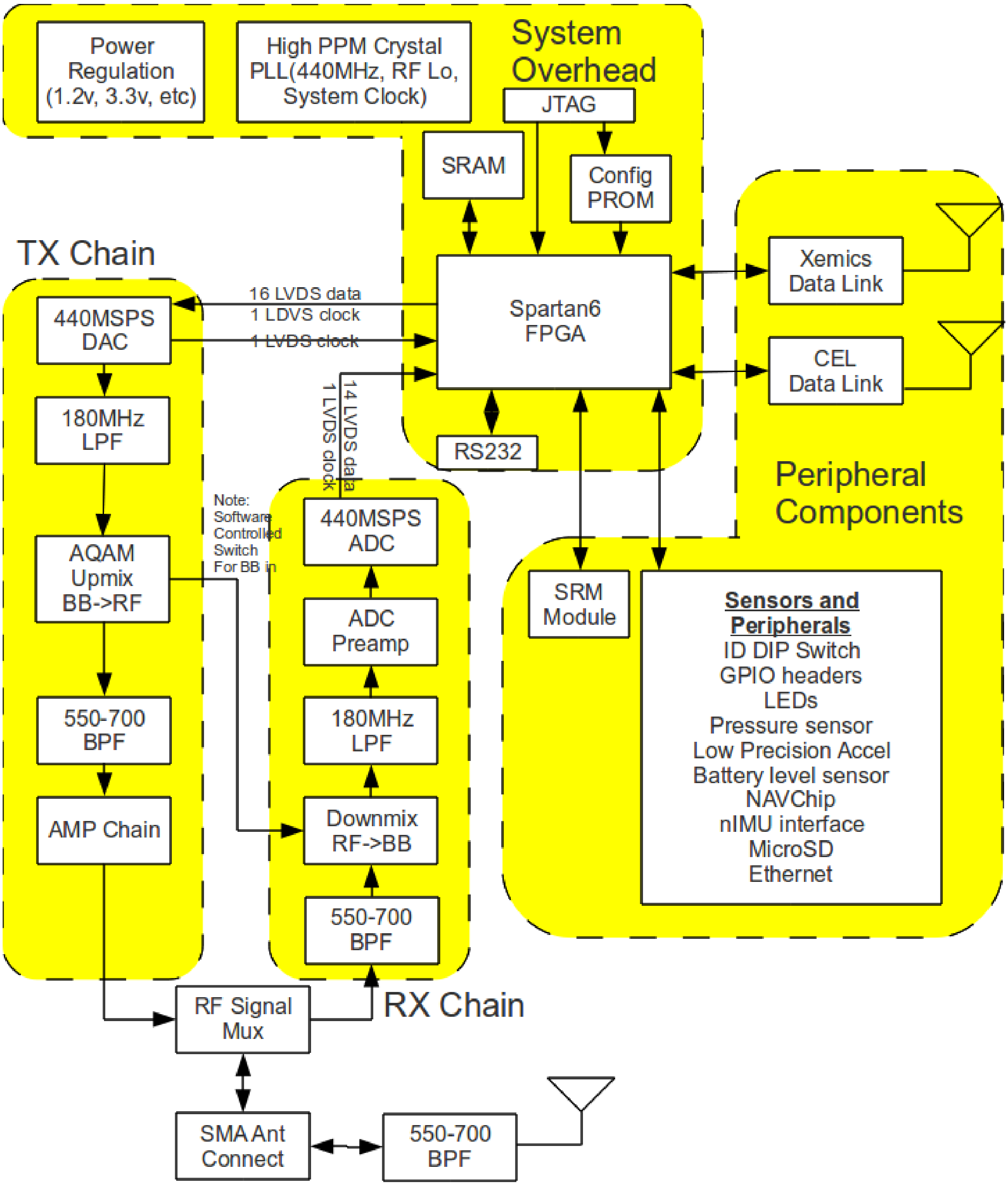


Figure 4.1: Revision 3 Component Level Block Diagram

4.1 FPGA Digital System

The digital system implemented on the FPGA is at the core of the revision 3 locator's operation. The FPGA digital blocks along with the Microblaze soft-core processor are what turn the individual hardware systems of the revision 3 locator into a cohesive device. The FPGA design is responsible for three tasks: high-speed digital data movement, interfacing and control of peripherals, and wireless digital communications.

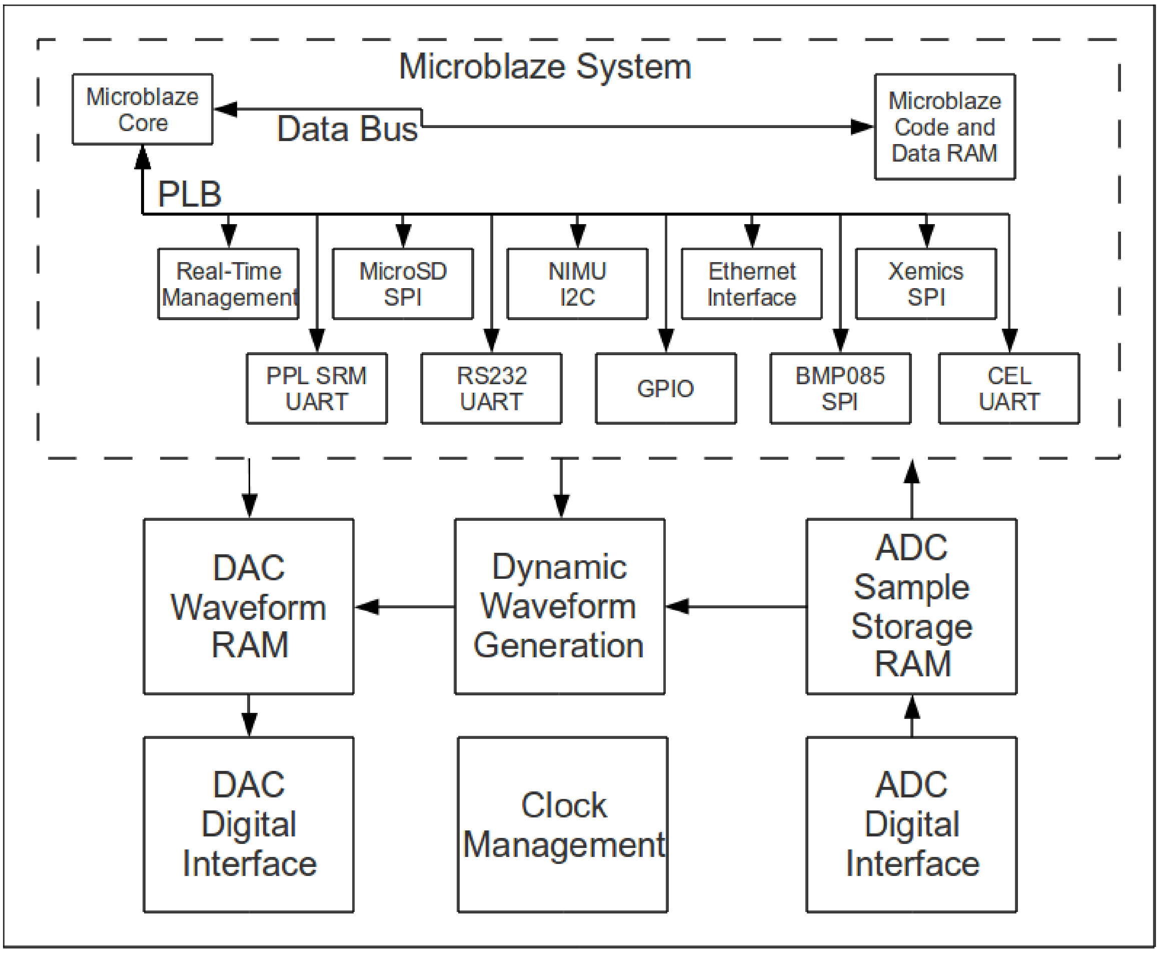


Figure 4.2: FPGA System Block Diagram

As seen in the FPGA system block diagram (Figure 4.2), several functional blocks are required to implement a basic TART system. Among these are the digital interfaces to the ADC and DAC, dynamic waveform generation, and the Microblaze system. The ADC and DAC functionality consists of storage for incoming and outgoing digital samples respectively, and the high speed digital interface for each of the two components. The dynamic waveform generation block is capable of creating I/Q waveforms for use by the DAC in real-time from either frequency or time domain input. This block is controlled by the Microblaze system, but is given direct memory access (DMA) to both the ADC and DAC sample RAM.

The digital interfaces to the ADC and DAC are described in the relevant sections later in this chapter. The dynamic waveform generation block is described after the ADC and DAC sections as knowledge of their operation is necessary to understand the dynamic waveform generation block. The Microblaze system is shown with the various communications blocks that are implemented within its architecture for this application. The architecture of the Microblaze system along with its implementation are described in the following subsection.

4.1.1 Microblaze

To coordinate system functionality, a microprocessor is implemented on the FPGA. The ability to use software as opposed to hardware for system functionality such as digital data communications and high-level hardware control provides an easier and more agile development flow during later PPL system testing and development. The processor used is the Xilinx 32-bit soft-core, which runs the C code that will dictate and integrate the operations of the various FPGA subsystems. From previous designs that include the PPL transceiver and the revision 2 mobile locator, a base of knowledge for the implementation of the Microblaze system was available.

Before the specific PPL application of the Microblaze system is described, a basic overview of this processor must be covered. Figure 4.3 shows the generic architecture of a Microblaze system. At the heart of the system is the Microblaze processor core. The processor is responsible for execution of code that is generated from the Xilinx-provided C language software tool chain. Code and data memory for the processor is implemented in the form of FPGA block RAM and is connected to the processor via a data bus. Separate from the data bus is a second bus, the processor local bus (PLB), that is used to connect various hardware functional blocks to the processor. Xilinx offers a wide range of PLB blocks for implementing functionality such as communications (UART, SPI, etc.), signal processing, and external memory interfacing. In addition to the Xilinx blocks, usergenerated PLB blocks can be written for custom functionality such as interfacing the Microblaze with HDL blocks external to the Microblaze architecture.

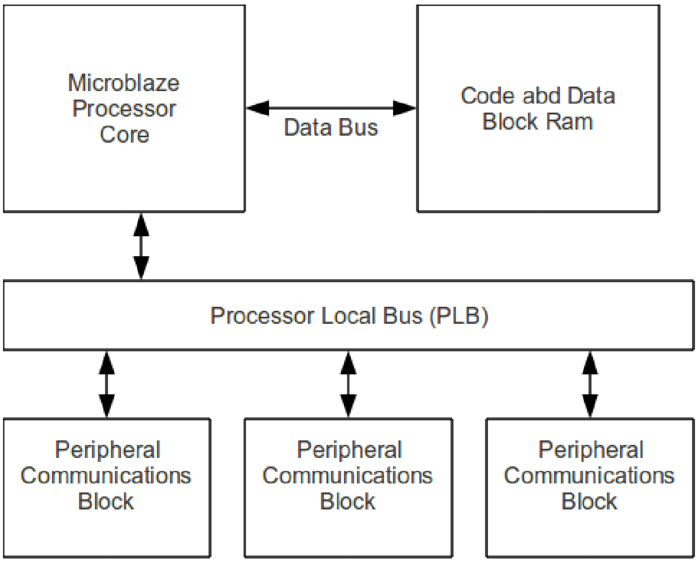

Microblaze Processor Core
Data Bus
Code abd Data Block Ram
Processor Local Bus (PLB)
Peripheral Communications Block
Peripheral Communications Block
Peripheral Communications Block

Figure 4.3: Generic Microblaze System Architecture

The Microblaze system is responsible for executing the code that manages the operation of all subsystems on the FPGA. In addition to this functionality, the Microblaze also contains standard communications interfaces for interfacing with smaller hardware peripherals such as the barometric sensor and the digital radios. Xilinx offers a series of hardware communications blocks for standard protocols such as SPI, I^2C, and UART. These blocks come with basic low-level drivers that speed the implementation of higher level drivers for interfacing with the various hardware peripherals.

In the previous designs, the Microblaze system was built on a custom real-time system, which operated with a main polling loop that handled all real-time functions as well as general system overhead such as digital data communications. Previous experimentation conducted with Xilinx's real-time operating system showed that this approach provided a higher level interface to various system functionality such as virtual memory, thread execution, and process management. However, the development cost of implementing and maintaining a system based on the Xilinx real-time operating system outweighed the advantages it offered. For this reason, and with the success of the PPL-implemented system in previous revisions, the PPL system remained the approach for the revision 3 software stack.

4.2 Data-Link Operation

In the previous two revisions of hardware, the system control has revolved mainly around the Xemics 915MHz data communications device using a PPL communications protocol. This protocol has served three common functions. First, the protocol is used for controlling each unit within the system. Functionality such as mode switching, device configuration, and transmission control are all implemented in a basic command protocol. Second, small-scale data transportation can be handled by the low-speed 915MHz data link. In previous revisions, this data has been restricted mainly to movement of physiological data from mobile locator devices. Finally, the 915MHz protocol was designed to provide a common time base on the order of microsecond accuracy through use of a system-wide timing packet. This synchronization is used as a time base not only for the communications protocol but also for various other system tasks such as transmit and receive timing windows.

4.2.1 Protocol Structure

The application for the protocol used over the 915MHz radio is targeted directly for PPL application. For that reason, the protocol is fairly rigid in terms of communications flow and structure. At its heart, the protocol has a rigid time-slotted time division multiplexing (TDM) structure.

The protocol is based on a frame structure. Within each frame, all units are allotted a time slot for transmission over a set period of time. A basic system configuration, consisting of a beacon, two reference units, and two mobile units, is shown in Figure 4.4. The reference units and mobile units directly correspond to the reference stations and mobile locators, respectively. The beacon unit is not an additional physical piece of hardware but rather functionality that is implemented at one of the reference units. One of the reference units is responsible for transmitting the initial synchronization packet in a frame.

Each frame begins with the beacon unit transmitting a synchronization packet. The reception time of the packet is used to set the timing synchronization among all other units in the system. Then the other units in the system each transmit during a predetermined time slot that is received by the beacon unit as well as by the reference units to provide greater coverage though redundancy. Each frame can be adjusted at design time for a somewhat arbitrary time length. There are restrictions on the time length, such as allowing enough time for each unit to transmit a meaningful amount of data and keeping it short enough to provide frequent synchronization packets. In previous implementations of this protocol, frame sizes of 1/4 second and 1 second have been used for various system applications.

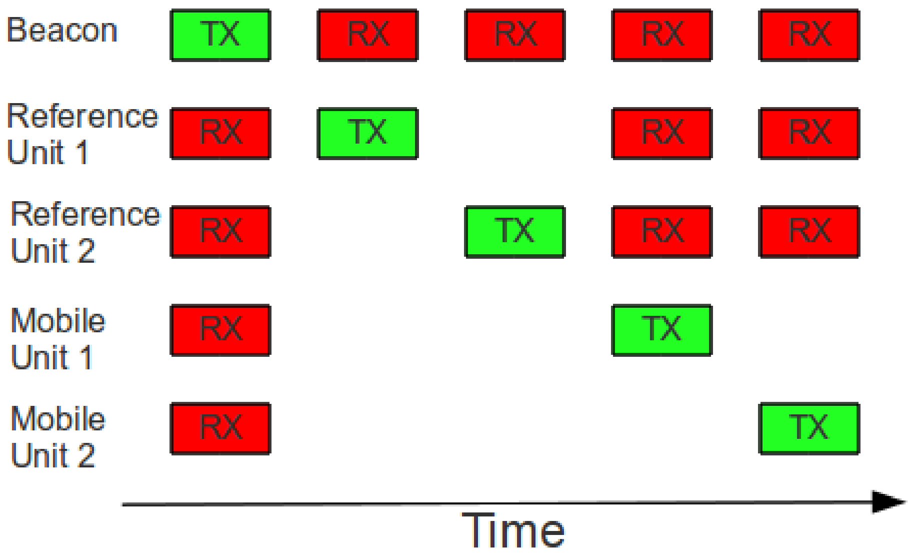


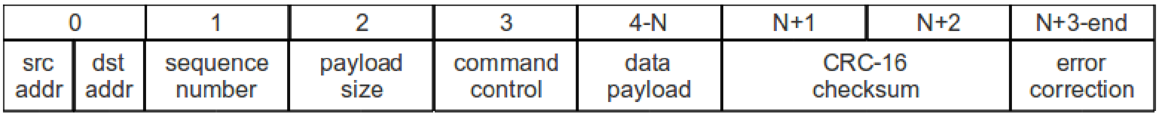
Figure 4.4: 915MHz TDM Protocol

In RF communications systems, there is the possibility of missed or corrupt data packets. This issue is dealt with through use of a packet acknowledgment system. Command packets are acknowledged upon receipt; if no acknowledgement is received, the data packet is resent.

Another major issue related to missed or corrupt data packets is the synchronization packet being missed. Each frame's timing is based on the start of the synchronization packet. If this packet is missed, the common time base for the TDM protocol is lost. To cope with this issue, the time base can be extrapolated from the last received synchronization packet. However, due to clock drift, eventually the extrapolated time base will drift far enough that it will move past the bounds of transmission guard times, causing collision issues. An analysis of the effects can be found in the design for the revision 1 mobile hardware[2].

4.2.2 Protocol Data Structure

The protocol data structure used in the revision 2 hardware was based around reliable transmission with support for generic data payloads. The structure of a generic packet is shown in Figure 4.5. The packet contains a source address and a destination address, each of which identifies a unit in the system's network. A sequence number is used for tracking duplicate packets. A payload length is included to indicate the size of the data payload. A control byte is included to indicate various protocol functions such as command acknowledgement. Following the command byte is a variable-length data payload field. At the end of each packet is a 2-byte CRC-16 checksum used to detect transmission errors. Finally, the end of the packet contains hamming data for forward error correction.



0		1	2	3	4-N	N+1	N+2	N+3-end
src addr	dst addr	sequence number	payload size	command control	data payload	CRC-16 checksum		error correction

Figure 4.5: 915MHz Data Packet Structure

The first version of this protocol was implemented in the revision 1 mobile locator hardware. The protocol as described above was the revised current state used with the revision 2 hardware. With the inclusion of the Xemics 915MHz device in the revision 3 hardware, it is likely that this protocol will be implemented for future field testing of the PPL system. Certain changes will need to be made during the testing of the revision 3 hardware for the protocol to fully suit the needs of the new system configuration; however, this protocol provides a solid implementation from which the revision 3 915MHz protocol will be built.

4.3 ADC Operation

The operation of the ADC in the system has several analog and digital considerations. The RF signal is received from the external antenna and then processed though an analog chain of preamplifiers and down mixers. After being sent through the ADC, the digital output is passed into the FPGA and processed to slow the signals to a speed the FPGA logic is capable of handling.

4.3.1 ADC Analog Path

The ADC analog receive path consists of several filters and mixers to condition the RF signal before it enters the ADC, as shown in Figure 4.6. The input band from the RF antenna that is required ranges from 550–700MHz.

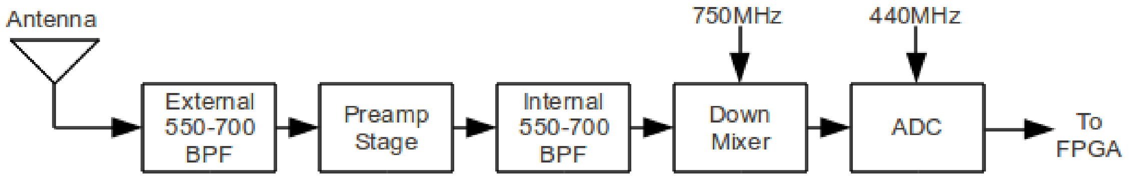


Figure 4.6: ADC Analog Path

The signal is received on an external antenna and immediately passed though a low-ripple fast-dropoff bandpass filter external to the locater device. The passband of this filter corresponds to the 550–700MHz band that is required. The signal is then passed on to the locator PCB and through a preamp chain. To remove any out-of-band noise that was increased by the preamp chain, another 550–700MHz band pass filter is used.

The now-conditioned signal is mixed with a local oscillator of 750MHz to bring it into base-band. The ADC is clocked at 440MHz, giving a Nyquist rate of 220MHz, or a DC to 220MHz baseband signal. The baseband signal from DC to 220MHz is a flipped version of the input RF signal from 530–750MHz. This is because an upper sideband image from the mixing process falls in the baseband range. Figure 4.7 shows the mixing process on an input RF signal that ranges from 600–700MHz. The ramp in amplitude as the frequency increases is used to explicitly illustrate the flipping effect caused by the mixing method. The highlighted portion of the post-mixed signal on the right of Figure 4.7 shows the flipped baseband image that is sampled by the ADC.

4.3.2 ADC Digital Interface

After the incoming RF signal has been processed and acquired by the ADC, it must be moved into the digital domain of the FPGA. The digital data bus from the ADC is configured as a 14-bit parallel low voltage differential signaling (LVDS) bus with a data rate of 440MSPS (220MHz

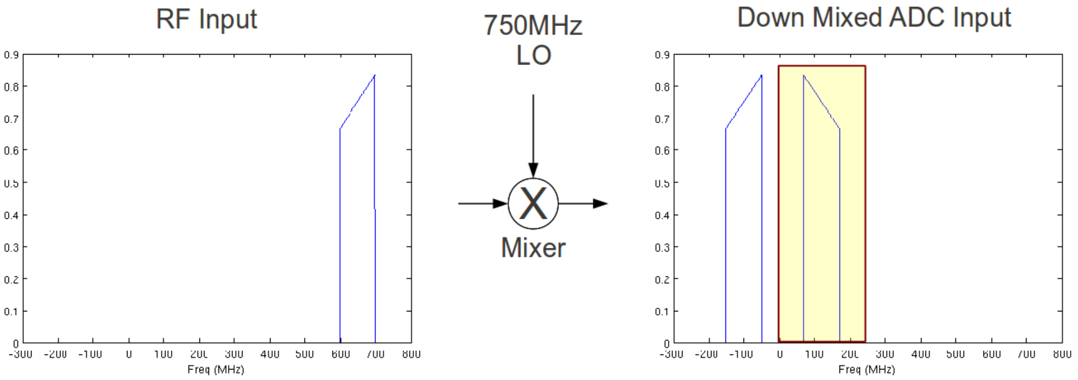


Figure 4.7: RF Signal Down Mixing

double data rate (DDR)). This bus is input to the FPGA and processed at the FPGA's input pads to condition the signal for use within the FPGA.

The ADC signals when input to the FPGA, as well any DDR signals fed into the FPGA, are converted to single data rate (SDR) signals at the input pads. All signals within the FPGA need to be SDR signals. Additionally, if the ADC samples were taken in at full speed, the FPGA would need to support 440MHz signals, which is well past the speed tolerance of the device. To cope with this timing constraint, a serial-to-parallel decoding system is used at the input pads to reduce the signal speed by increasing the bus width.

After LVDS termination and conversion to SDR, each of the input signals to the FPGA is run though a Xilinx SERDES serial-to-parallel block at the input pad (Figure 4.8). This serial-to-parallel block buffers four serial input bits and outputs a 4-bit bus at quarter the input data rate. Through this method, rather than a single ADC sample being moved on the FPGA at 440MHz, a 4-sample-wide bus is produced operating at 110MHz. This speed reduction allows this design to meet the timing constraints the FPGA device. Figure 4.9 shows a timing diagram for ADC sample input to the FPGA. Each DDR data transition represents a new 14-bit ADC sample, and each SDR data transition represents a new 4 ADC sample, 56-bit wide bus to the FPGA logic.

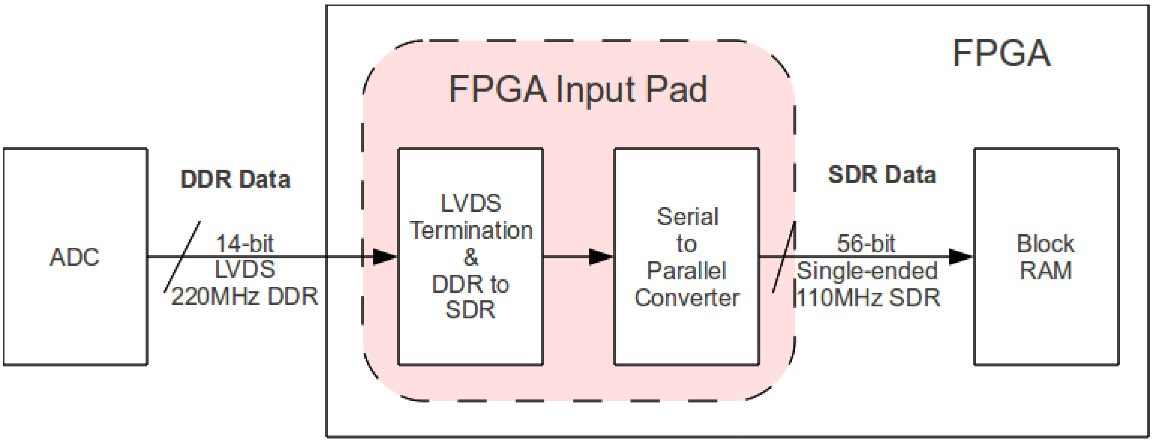


Figure 4.8: ADC FPGA Interface

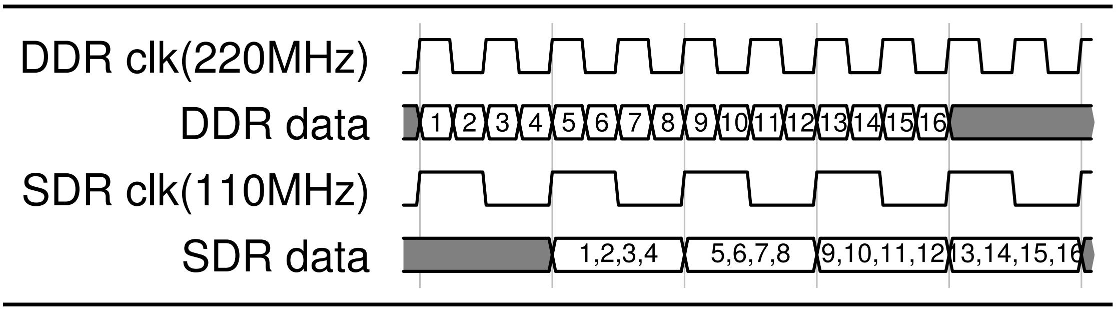


Figure 4.9: ADC FPGA Interface – Digital Timing Diagram

4.4 DAC Operation

The operation of the I/Q DAC differs significantly from the DAC used in previous revisions. Using I/Q modulation brings with it several key differences in how transmitted signal waveforms are generated and moved to the DAC. In addition to the differences I/Q modulation brings, there are also many new considerations with the DAC analog and digital paths. The new DAC provides a host of features and operation characteristics related to digital processing of the waveform within the DAC device itself.

4.4.1 I/Q Modulation Basics

I/Q modulation is a step away from the signal-generation methods used in previous revisions of the hardware. Previously, a single DAC was run at a 440MHz sample rate, and a flipped baseband image of the final signal was produced. This method provides slightly under 220MHz of usable bandwidth. After the baseband signal is produced, it is mixed with a 750MHz carrier to produce the final RF output (Figure 4.10).

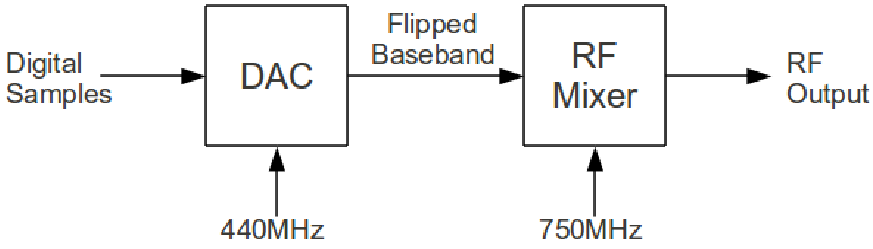


Figure 4.10: Theory of RF Operation, Previous Revision

There are several shortcomings to this setup. The two most troubling issues are LO leakage and image leakage. Both issues increased the complexity required to meet FCC compliance; in both cases, bandwidth had to be sacrificed in order to achieve FCC compliance (these issues were discussed in Section 3.3.1).

In the new hardware, an interpolating I/Q DAC is used, which helps fix the above-mentioned problems. Before any discussion of how these problems are fixed with the new DAC, a background in I/Q modulation theory is needed.

Figure 4.11 shows the typical signal chain for the I/Q DAC, an Analog Devices AD9122, from the data sheet[4]. I/Q modulation uses analog quadrature modulation on two separate signals to produce the final RF output. There are varying methods of I/Q signal generation and the analog quadrature modulation that can be used to produce the final RF signal, several of which are supported by the AD9122; this section, however, focuses on the methods used in this hardware.

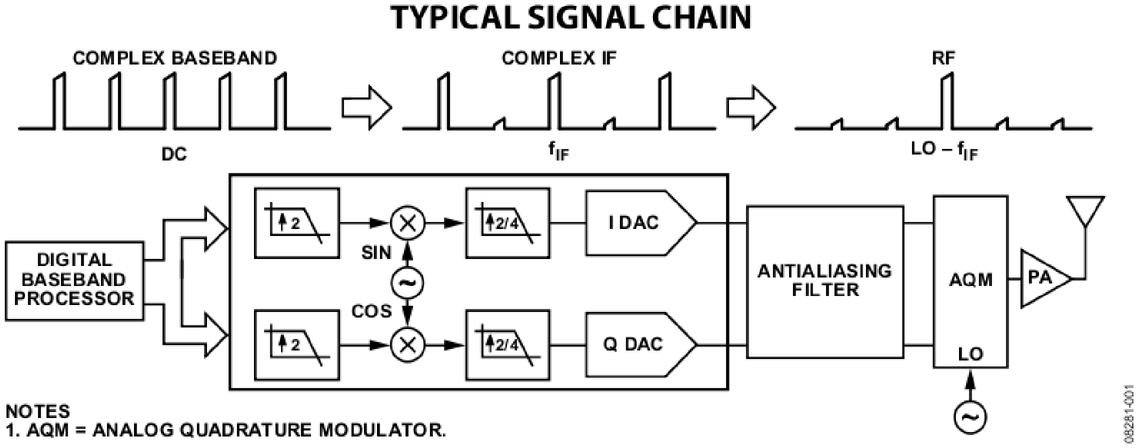


Figure 4.11: AD9122 IQ DAC Theory of Operation[4]

Using the two input signals [the in phase (I) and the quadrature phase (Q)] at half the intended Nyquist rate, it is possible to recombine them and produce the final signal with the full bandwidth. For instance, in the PPL system, the needed usable bandwidth is 150MHz, which would require a sampling rate of at least 300MHz to produce the signal. In the PPL system, a sampling rate of 440MHz is used because extra bandwidth is needed on the higher and the lower sides of the signal to compensate for filtering and imaging issues. To produce this 220MHz of bandwidth, the I and Q signals each will have only 110MHz of bandwidth, each with a sample rate of 220MHz. Examples of I and Q waveforms for a PPL multicarrier signal can be seen in Figures 4.12 and 4.13.

4.4.2 DAC Digital Data Path

With the interpolating I/Q DAC, the digital data goes though several stages of digital conditioning before entering the analog domain. The layout of the DAC's digital data path can be seen in Figure 4.14. At the beginning of the digital data path is a premodulation stage. Next are modulators,

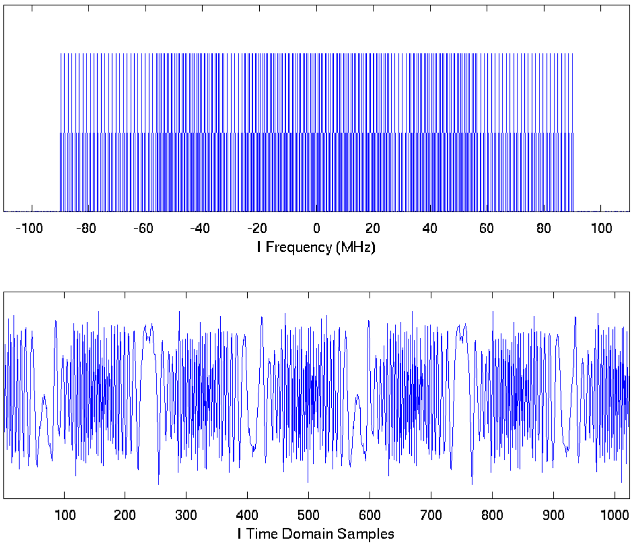


Figure 4.12: PPL Multicarrier Waveform – I Component

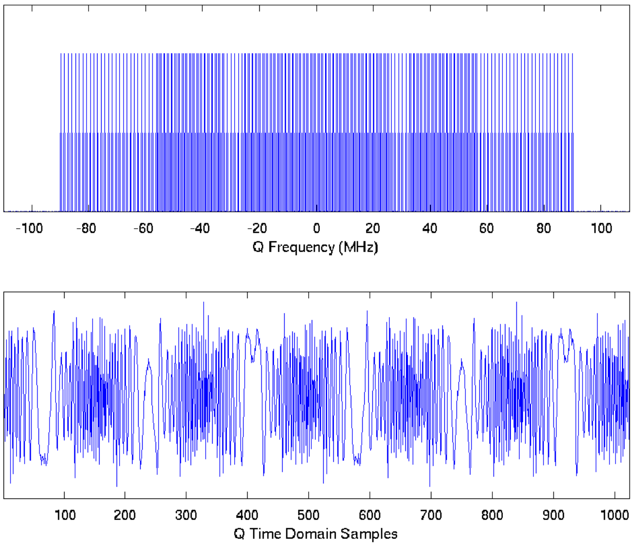


Figure 4.13: PPL Multicarrier Waveform – Q Component

interpolaters, and filters implemented in three half-band filter stages (HB1, HB2, and HB3 in Figure 4.14). Between HB1 and HB2 is an optional modulation stage not used in this application. Finally, at the end of the path, there are a phase and offset correction unit (not used in this application) and a sinc compensation.

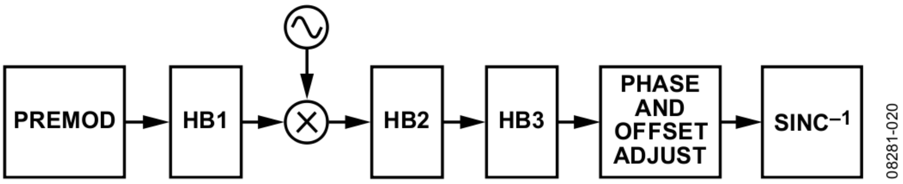


Figure 4.14: AD9122 Digital Data Path[4]

Possibly the most noteworthy parts of the DAC digital data path are the premodulator and the half-band filters. The premodulator can be used to modulate the signal, shifting it by $F_{in}/2$. This allows a signal shift that can be used in the half-band filters to relocate the output signal. The three half-band filters each consist of a 2x interpolation and digital filter stage. Each of the three half-band filters can be bypassed or used in several modes. If the half band-filter is used, it will apply 2x interpolation through means of zero interleaving the time domain signal by a factor of 2. The output of the filter stage will be at twice the sample rate and bandwidth. An artifact of the zero padding interpolation is the occurrence of images in the newly expanded bandwidth, as shown in Figure 4.15.

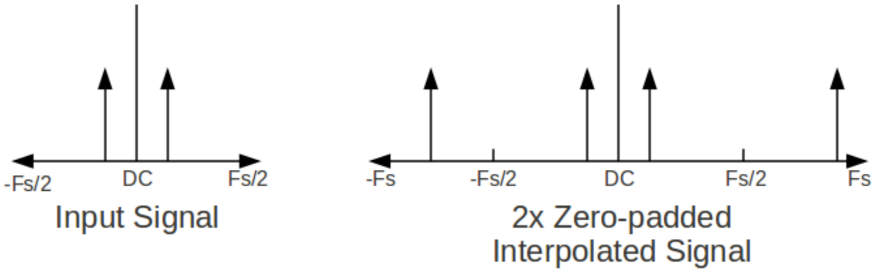


Figure 4.15: 2x Interpolation Imaging

Through selective filtering, unwanted images are removed, and the possibility of effectively modulating the signal also exists by choosing one of the higher images. Various filters are available at each half-band stage, as shown in Figure 4.16. Each filter reduces the usable bandwidth at output. In this application, only HB1 and HB2 are used, bypassing HB3, which results in a 4x interpolation factor and a usable bandwidth of approximately 180MHz. Although there is a loss of usable bandwidth, the requirements of the PPL project need only approximately 150MHz of usable bandwidth.

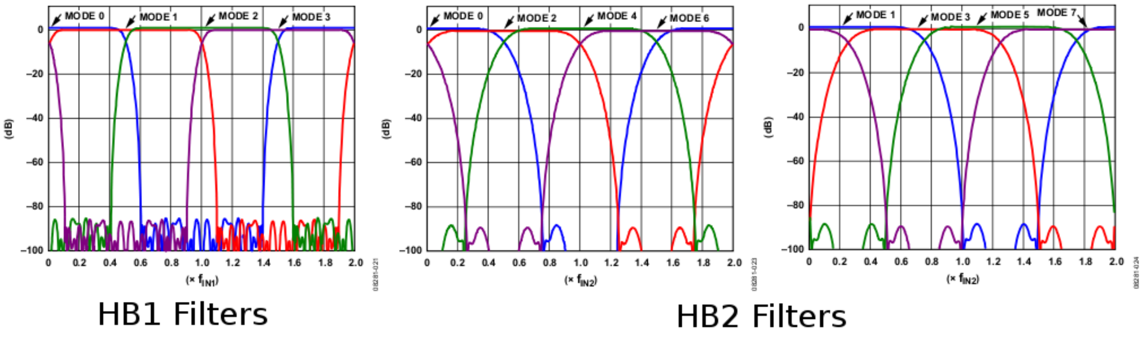


Figure 4.16: Half-Band Filters of the AD9122[4]

To better understand the particular mode in which the AD9122 DAC is run for this application, it is helpful to follow the life of a signal though the digital data path. Rather then following the I and Q components of the signal separately, it is easier to view the composite complex signal they represent. The DAC takes in a complex signal with a bandwidth from –110MHz to 110MHz, as seen in Figure 4.17.

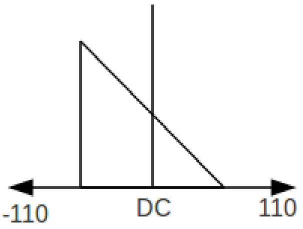


Figure 4.17: Complex Input to DAC

The first stage of digital conditioning applied to the signal is a shift of $F_{in}/2$ by the premodulator

(Figure 4.18).

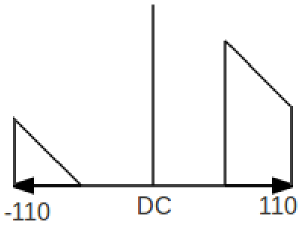


Figure 4.18: Modulated Signal

At this point, the signal is run though the HB1 stage. 2x interpolation is applied, doubling the number of samples and the bandwidth, and a filter is applied to remove the unwanted images. The output of this stage is seen in Figure 4.19. The filter used is shown as a red curve that retains 80% of the original F_{in} (176MHz total bandwidth) centered about -110MHz.

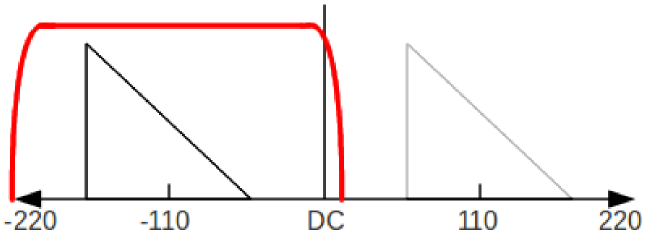


Figure 4.19: Output of HB1

The output of HB1, now interpolated and filtered, is fed into HB2. In HB2, another 2x interpolation is applied, for a cascaded total of 4x interpolation compared with the input. Again, the interpolated signal is filtered to remove unwanted images. This time, the filter covers a full 440MHz, with no loss in usable bandwidth (Figure 4.20).

The HB3 block is bypassed and has no effect on the signal. At this point, the resulting I and Q data streams are fed into the DAC section of the AD9122. The analog I and Q signals are output from the AD9122 chip and are ready to enter the analog modulation and conditioning chain.

Sinc compensation is used in this application. The need for sinc compensation comes from the frequency response of DACs in general. Figure 4.21 shows an attempt to output a flat frequency

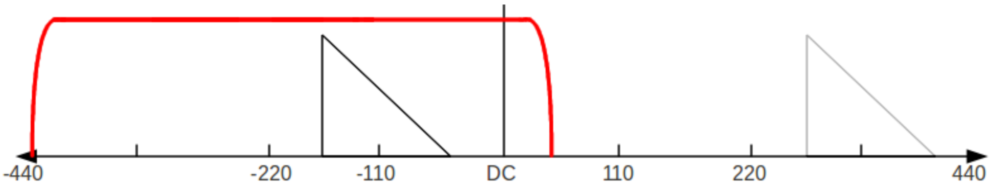


Figure 4.20: Output of HB2

signal from a DAC without sinc compensation. The sinc envelope shown is a common artifact of DACs and alters the output to differ from what was intended. In the first Nyquist zone from DC to $F_s/2$, a distinct curve can be seen in the signal that was intended to be flat. The AD9122 offers sinc compensation at the end of the digital data path. When enabled, the signal will be digitally altered to counter this effect. In this application, the DAC sinc compensation is used to remove the computational burden from whichever PPL signal generation method is used.

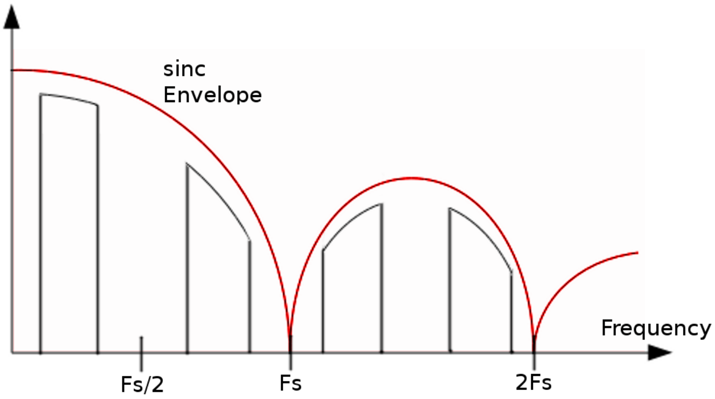


Figure 4.21: DAC Output Sinc Envelope

4.4.3 DAC Analog Path

The analog path for the DAC signal is somewhat simpler in theory than the ADC analog chain. An overview of the DAC analog path can be seen in Figure 4.22. The initial analog output from the DAC are two signals: the in phase signal and the quadrature phase signal. These baseband signals, which range from DC to 220MHz, are input to an analog quadrature modulator (AQM). The theory of operation for the AQM can be seen in Figure 4.23. The RF output from the AQM is band pass filtered, amplified, band pass filtered again, then fed to antenna.

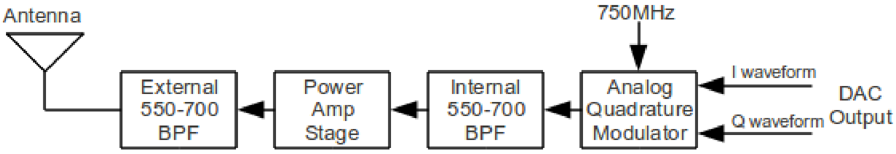


Figure 4.22: DAC Analog Path

The AQM operates by modulating the in phase component with a 750MHz sine wave (0°). This takes the baseband signal shown in Figure 4.20 and shifts it to the RF output range of 530–750MHz. The quadrature phase signal is similarly modulated; however, it is modulated with a cosine (90°). The modulated output from the I and Q waveforms are summed and produce the final RF output.

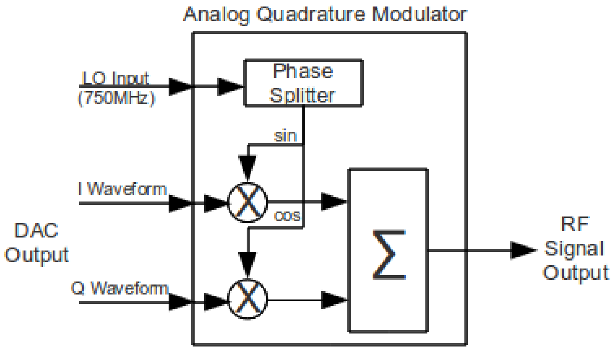


Figure 4.23: AQM Theory of Operation

4.5 Dynamic Waveform Generation

The decision was made to provide the ability for the locator to generate arbitrary I/Q waveform pairs in real-time from both time domain and frequency domain representations. There are several applications of this real-time dynamic waveform generation, the most interesting and challenging being its use in TART's transactional synchronization. In addition, this real-time waveform generation could be used in various testing applications as well as allow for generation of waveforms from frequency domain representations of multicarrier waveforms.

4.5.1 I/Q Waveform Generation

Before the intricacies of TART's synchronization are tackled, the basic real-time waveform generation must be implemented. As discussed in Section 4.4, the DAC used in this design is of an interpolating I/Q type. Unlike a standard DAC, this I/Q DAC does not use standard time domain signals as digital input. Rather, two signals are processed by a pair of internal DACs, which are then modulated using an analog quadrature modulator for the final RF output. The basics of how I/Q modulation works, as well as the specifics for the AD9122, were covered in Section 4.4.

I/Q Signal Generation Concept

This subsection focuses on the methods used to generate the I/Q waveforms to produce an arbitrary RF output in real time. Several Xilinx hardware blocks are used to generate the I/Q output, including but not limited to the Xilinx FFT implementation.

The goal of the the real-time generation block is to produce the I/Q waveforms from an arbitrary frequency definition in a time span of a few milliseconds. The conceptual data flow is shown in Figure 4.24 as a block diagram.

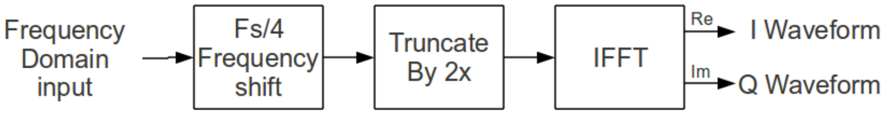


Figure 4.24: Conceptual Block Diagram Real-Time Signal Generation

The input to the generation block is a frequency domain representation of the final DAC output represented by discrete frequency bins. The input is conditioned first with a shift of $F_s/4$. The signal can be shifted to select either the negative or the positive image, effectively applying a flip of the signal in the frequency domain or not. After this shift, the number of frequency bins is reduced by 50%. At this point, a representation of the intended output is truncated from $-F_s/4$ to $F_s/4$. The new signal is no longer real but rather an imaginary representation of the I and Q signals. After being run through an IFFT, the output corresponds to the time domain I and Q signals. The real output is the in phase signal, and the imaginary is the quadrature.

To illustrate this process, an arbitrary signal will be run through these steps. Figure 4.25 shows a real frequency domain signal that is input to the I/Q generation process. The content from DC to $F_s/2$ represents the intended output of an I/Q DAC. Figure 4.26 shows this signal with the $F_s/4$ shift and 50% truncation applied. Finally, Figure 4.27 shows the time domain output of the IFFT, the I and Q time domain signals.

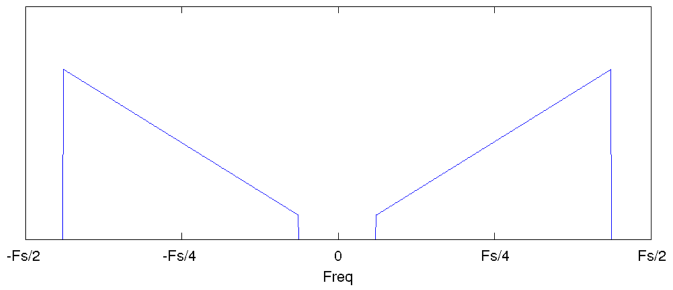


Figure 4.25: Frequency Domain Input to I/Q Generation

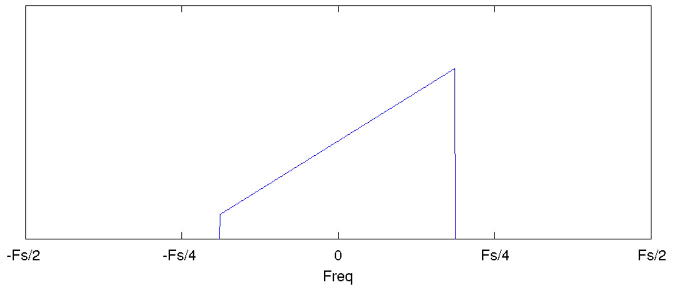
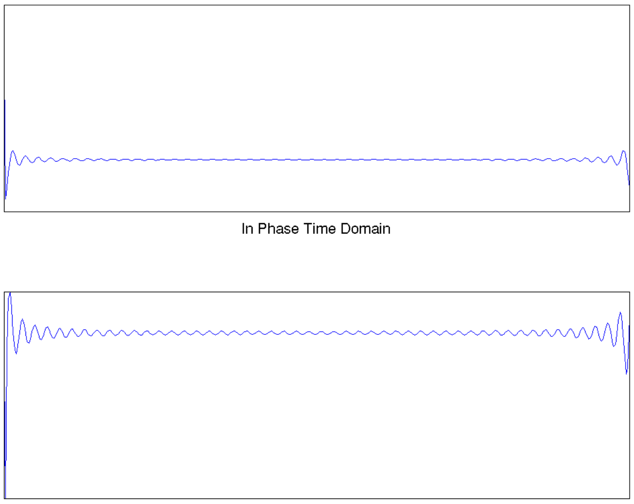


Figure 4.26: Shifted and Truncated Signal – Frequency Domain

4.5.2 I/Q Dynamic Waveform Generation Implementation

With the basic concept of I/Q signal generation understood, the hardware implementation can now be discussed. This signal generation is implemented in hardware using VHDL and Xilinx IPCores.



In Phase Time Domain

Quadrature Time Domain

Figure 4.27: Time Domain I and Q Waveform Output of IFFT

The decision to implement this signal generation in hardware instead of in software was made for several reasons. First, a hardware implementation will offer the greatest speed, because it can be clocked faster then a Microblaze and it operates as a parallel pipeline. Additionally, with this task offloaded to dedicated hardware, the Microblaze is freed up to perform other software tasks, which is crucial in time-sensitive systems such as PPL. The downside to a hardware implementation is that more effort and time are needed during development and testing. Also, after implementation, more effort is needed to make changes compared with a microprocessor software implementation. After the pros and cons of each implementation are weighed, the hardware is a clear winner due to the real-time applications in PPL.

The true hardware implementation is based on the concepts discussed in Section 4.5.1. Changes have been made to this configuration to account for the nature of real-hardware implementation. In addition, some extra processing is added to account for the specific needs of the PPL project. The hardware block diagram can be seen in Figure 4.28.

The block was designed with flexibility to support future applications. In addition to generating an I/Q waveform pair from a frequency definition, it has the ability to take baseband time domain

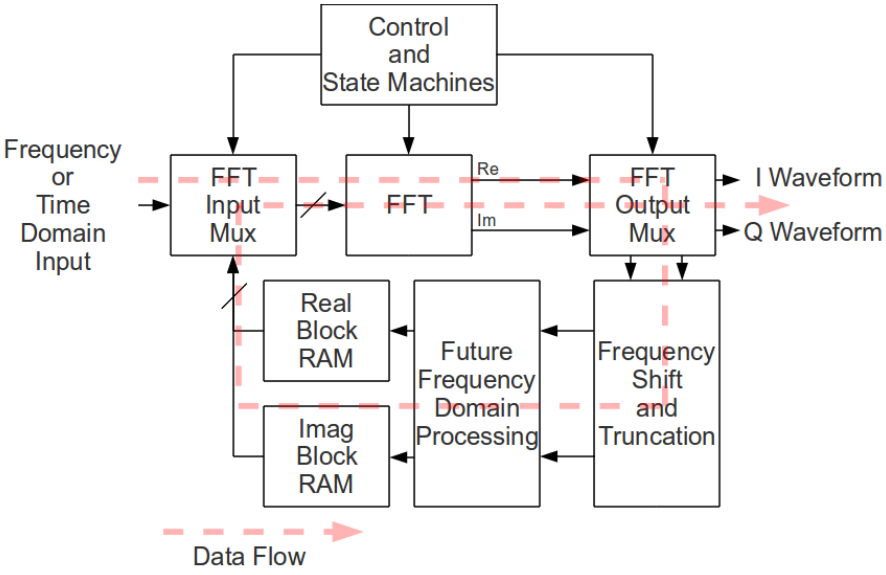


Figure 4.28: Real-Time I/Q Waveform Generation Hardware Block Diagram

input from the ADC and produce an I/Q waveform pair.

Initially, a frequency or time domain signal is input to the generation block. If a time domain signal is used, it is put through an FFT to obtain a frequency domain representation. The frequency information is shifted and truncated to the non-real I/Q frequency representation. At that point, further frequency domain processing can be injected, some of which is discussed in Section 4.5.3. The FFT output is in non-natural ordering and is stored in RAM until the FFT calculation has completed. Upon FFT completion and with processed non-real I/Q frequency data stored in RAM, the FFT block is dynamically reconfigured to perform an IFFT of half-length. The IFFT is performed on the data stored in RAM. The output stream from the IFFT is the in phase (real) and quadrature (imaginary) time domain signals that are ready for input to the DAC or further time domain processing.

The FFT used in this design is the Xilinx's IPCore[21]. This IPCore is highly configurable, and the mode of operation was chosen carefully to suit this application. Most of the configuration options of the IPCore relate to resource usage versus performance trade-offs. In this application, performance is the more important of the two factors, but both must be considered. The configuration with the least latency available is a pipelined streaming architecture. The trade-off of resource usage was minimal compared with the gains in latency. Latencies of about 2x smaller, depending on FFT size, are possible with pipelined architecture compared with a Radix Burst implementation. The resource trade-offs amount to less than 1% of the total Spartan 6, which can be considered almost negligible in this application.

Other configuration options for the IPCore were also changed from default settings. A run-time variable length option has been selected to accommodate this design. When time domain information is used as input, a forward FFT must be run on the data to produce the needed frequency domain data. If a time domain waveform of 2048 samples is input to the forward FFT, the output will produce a 2048 bin frequency representation. After the $F_s/4$ frequency shift and 2x truncation, the data will be only a 1024 bin frequency representation. The data then needs to be run through a 1024 length IFFT to produce the I/Q time domain waveforms. Because of this need, the FFT block is run-time configurable to support multiple power-of-2 FFT sizes. Additionally, with future extension of the control and state machine, this block could be extended to support arbitrary power-of-2 length time and frequency domain input.

With this core of the block implemented, basic testing was done through HDL simulations in ModelSim. The first step of the testing was to verify the correct operation of the signal generation block. For this purpose, a VHDL test bench was written to feed simulated ADC signals from MATLAB into the simulation. The output of the simulation was compared against a MATLAB model of the process to verify the proper function of the block.

The input to the simulations was a sinusoid with zero degree phase at one bin past $F_s/4$ making use of 90% of the 14-bit dynamic range. A single sinusoid was chosen for test input due to the ease of it being tracked through the system. The expected output for this system would be low frequency sinusoids for both the I and the Q time domain waveforms. As can be seen in the output in Figures 4.29 and 4.30, the MATLAB and ModelSim outputs track each other perfectly. Although they are not equal in amplitude, they are simply linear scalings of each other. This is due to the fact that the simulation in MATLAB was run at double precision, while the hardware implementation is run with truncating fixed point arithmetic.

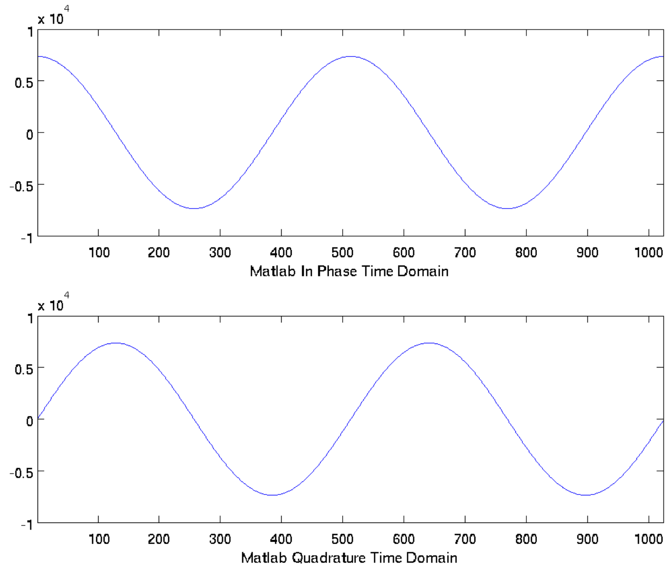


Figure 4.29: MATLAB I/Q Generation Simulation Output

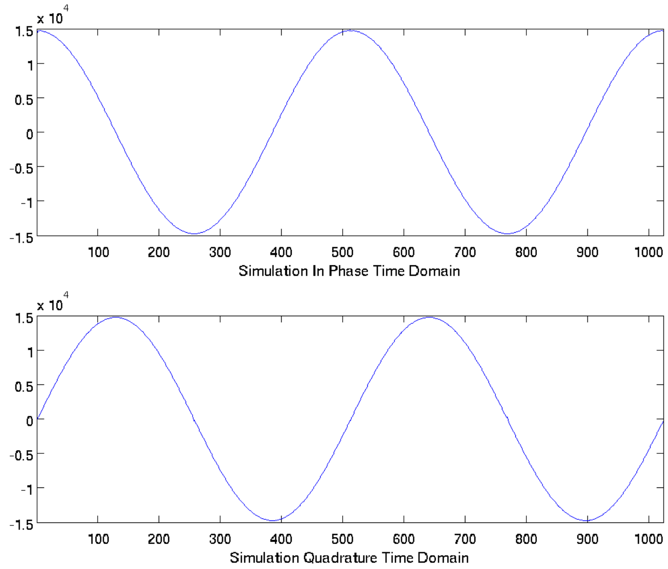


Figure 4.30: ModelSim I/Q Generation Simulation Output

Future Work

Although the dynamic waveform generation block has been implemented and tested in simulation, there are still outstanding issues and features that must be added to the block to realize its full potential. Two of these issues include signal scaling for full dynamic range usage and frequency bin masking for removal of noise when ADC data is used as input.

The first issue of dynamic range usage is simple in theory. The time domain I and Q signals output from the block will not necessarily be scaled to utilize the full 16-bit dynamic range available. To make use of all the transmission power available, the time domain signals should be scaled in each case to make almost full use of the available dynamic range. This issue must be addressed once time domain signals have been produced and cannot be fixed in the frequency domain input to the block.

To address this issue, a simple scaling value needs to be applied to all samples of the time domain signals to maximize dynamic range usage. The implementation of this method in real time on the FPGA has certain complications that need to be documented. First, the scaler value needs to be obtained. The scaler value is simply a ratio of the largest amplitude representable by 16 bits to the largest amplitude time domain sample (Equation 4.1).

$$scale = \frac{\text{16bit}_{max}}{\text{sample}_{max}} \tag{4.1}$$

One issue with obtaining this value is that all time domain samples must be examined before the value can be found, which means that all samples must be buffered and stored as they stream from the generation block before scaling can occur. This will introduce additional delay into the generation block. However, the delay will be on an order of magnitude that will not affect the PPL system adversely. Equation 4.2 shows the delay that would be introduced in finding this value when operating at 110MHz and using a 2048 sample signal.

$$\frac{1}{\text{110MHz}} * 2048 = 18.6\mu\text{S} \tag{4.2}$$

The second issue with the dynamic range scaling is the hardware implementation of the scaling mathematics. Until this point, in the dynamic generation block all mathematic operations have been performed with integer data types. When scaling by a non-integer value, either a fixed-point or floating-point multiplication method must be used. Both implementations have merit. A fixed-point method provides speed and a simple implementation. The floating-point method provides higher accuracy at the cost of more complicated implementation. For the purposes of this project, the small single-bit inaccuracies that would be introduced with the use of fixed-point math are inconsequential. With the large number of hardware multipliers available on the Spartan 6 device, a parallel scaling block could be added that would introduce time delays on the order of single microseconds.

The issue of frequency bin masking also needs to be addressed in the future. Frequency bin masking would be used when the dynamic waveform generation block is used to process real-world ADC data. When producing a waveform from ADC data, there is usually a known set of frequency bins that are of interest. To help remove unwanted noise from the signal, all but the frequency bins of interest would be set to an amplitude of zero directly prior to performing the inverse FFT. Figure 4.28 shows a block for future frequency domain processing. It is at this location that the bin masking will occur.

To remain flexible, the mask should be implemented as a programmable bit array. The bit array is of the same length as the size of the FFT used for signal generation. Each bit corresponds to a frequency bin, and its value determines whether the bin should be zeroed. By using a programmable bit array, the bins for zeroing can be changed on the fly from the base station over the control protocol.

4.5.3 TART Ranging Signal Data Integration

A promising use for the dynamic waveform generation block is integration of the TART ranging signal data into the multicarrier waveform. Data throughput has been a challenging issue to overcome with the TART location method. For TART to function, the base station processing computer must have the ranging signal from both sides of the two-way transaction. One straightforward approach is to have the mobile locator send frequency domain or time domain digital information back to the base station (Figure 4.31).

This method of retrieving the ranging signal data from the mobile unit has several issues. First, it

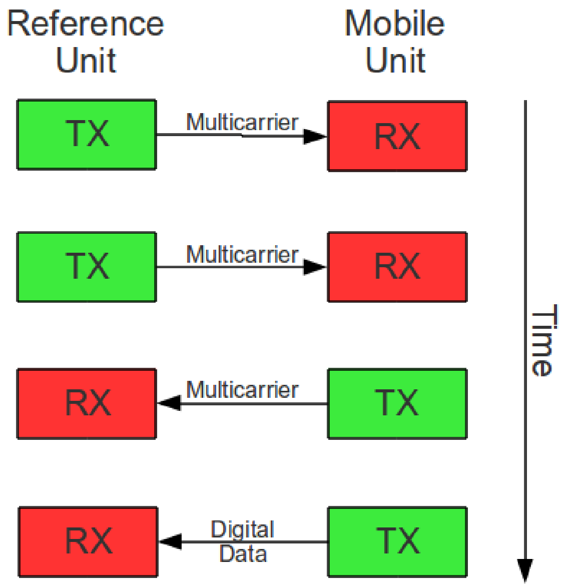


Figure 4.31: TART Transaction With Additional Digital Data Transfer

requires a large amount of reliable digital data bandwidth. Additionally, this method of transmission has issues as the number of mobile locators scales up. The bandwidth requirements become a hindrance extremely quickly as the number of mobile locators increases. To address this issue directly, a high-speed data link was included in the revision 3 hardware, but other options of addressing ranging signal data retrieval are also considered.

Through use of the dynamic waveform generation block, the needed data can be encoded into the response multicarrier waveform from the mobile unit to the reference units. Equations 4.4 and 4.5 show mathematical representations of the two signals that are key to TART's operation: the received signal at the mobile locator q and the reference antenna p, respectively. Equation 4.3 defines the entrywise, Hadamard or Schur product operator, $\circ$, used in this section.

$$\mathbf{Z} = \mathbf{X} \circ \mathbf{Y}, (\mathbf{Z})_{a,b} = (\mathbf{X})_{a,b}(\mathbf{Y})_{a,b} \tag{4.3}$$

$$\mathbf{r}_q = \mathbf{x} \circ \mathbf{h}_{q,p} \circ e^{-j\omega\tilde{\tau}_p}, \text{ and} \tag{4.4}$$

$$\mathbf{r}_p = \mathbf{x} \circ \mathbf{h}_{q,p} \circ e^{-j-\omega\tilde{\tau}_p} \tag{4.5}$$

In Equations 4.4 and 4.5, $\mathbf{x}$ represents the transmitted multicarrier waveform, $\mathbf{h}_{q,p}$ is the RF channel response between the mobile locator and the reference antenna, and the final term represents a random time offset that must be solved for in order for TART to operate.

The initial approach for retrieving this data was to have the mobile locator hardware digitize its received signal, $\mathbf{r}_q$, and transmit it back to the base station via digital data link. Although this method will work, there are several issues with its final implementation, as mentioned above.

Now consider the mobile locator's ability to generate multicarrier waveforms dynamically. A new signal, $\mathbf{x}^*$, is generated by inverting each value of the received signal vector.

$$\mathbf{x}^* = \frac{[\mathbf{1}]}{[\mathbf{r}_q]} = \frac{[\mathbf{1}]}{[\mathbf{x} \circ \mathbf{h}_{q,p}]} \circ e^{j\omega\tilde{\tau}_p} \tag{4.6}$$

This signal is then transmitted by the mobile unit to the reference antenna in place of the original multicarrier waveform. The reference antenna will receive a new signal, r_p^* (Equation 4.7).

$$\begin{aligned} \mathbf{r}_p^* &= \mathbf{x}^* \circ \mathbf{h}_{q,p} \circ e^{-j-\omega\tilde{\tau}_p} \\ &= \frac{[\mathbf{1}]}{[\mathbf{x} \circ \mathbf{h}_{q,p}]} \circ e^{j\omega\tilde{\tau}_p} \circ \mathbf{h}_{q,p} \circ e^{-j-\omega\tilde{\tau}_p} \end{aligned} \tag{4.7}$$

The received signal r_p^* can then be multiplied by the known multicarrier waveform $\mathbf{x}$ to produce a complex exponential (Equation 4.8). This complex exponential can be used to estimate $\tilde{\tau}_p$, which is the information that would be provided by returning a digital copy of the original received signal, $\mathbf{r}_p$.[3]

$$\begin{aligned} \mathbf{r}_p^* \circ \mathbf{x} &= \frac{[\mathbf{1}]}{[\mathbf{x}]} \circ e^{j2\omega\tilde{\tau}_p} \circ \mathbf{x} \\ &= e^{j2\omega\tilde{\tau}_p} \end{aligned} \tag{4.8}$$

The implementation of this method in hardware would only be a simple extension of the already implemented dynamic waveform generation block. Consideration would have to be given to the fact that precise and timely division operations would need to be implemented to produce the inverse signal shown in Equation 4.6. Using the many available DSP blocks of the Spartan 6 FPGA, an efficient pipelined architecture could be added to the dynamic signal generation block that introduces a negligible amount of delay.

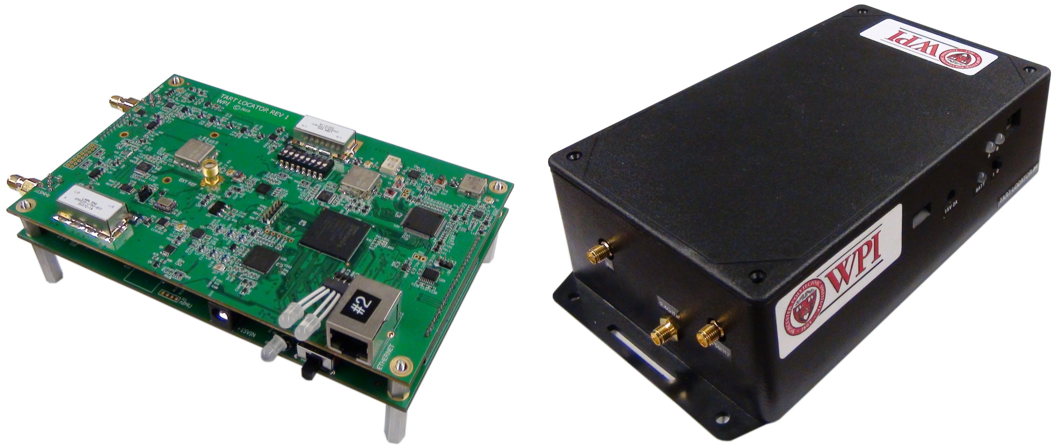
There is one minor disadvantage to this approach: in creating the multicarrier waveform $\mathbf{x}$, an effect know as the crest factor must be accounted for. The crest factor is a measure of peak-to-average ratio. The crest factor is optimized for our multicarrier signal by producing optimal phases for each of the individual sinusoidal carriers. When the received signal $\mathbf{r}_p$ is used to generate the new transmitted signal $\mathbf{r}_p^*$, frequency-independent phase offsets for each carrier cause the signal to no longer have the optimized crest factor. This results in $\mathbf{r}_p^*$ having a lower total transmitted power compared to $\mathbf{x}$ that is a function of the unknown channel response $\mathbf{h}_{q,p}$. This effect will have to be observed once this new method is implemented and tested in the field. However, it is likely that the advantages of this method will outweigh this negative effect and will outperform the use of a digital data link in both scalability and reliability.

With the revision 3 hardware system architecture designed and implemented, the ground work for a field-ready TART device has been laid. PCBs were designed and fabricated to support this design. Upon the arrival of the populated PCBs, preliminary system testing was conducted to verify the hardware design as well as the system design and associated FPGA blocks documented above. Chapter 5 documents the testing conducted immediately following initial power-on of the new hardware.

Chapter 5

Preliminary System Testing

Once the new locator hardware was received, preliminary system testing was conducted to verify the functionality of each component and subsystem. Figure 5.1 shows the new locator hardware as a PCB stack and assembled in an enclosure. The testing started with basic power-on tests and verification of each power rail and making sure current consumption did not indicate any circuit shorts. Following this verification, each subsystem, starting with the most basic such as the FPGA and clocking systems, was tested and verified for proper operation.



(a) PCB Stack (b) Assembled Enclosure

Figure 5.1: Revision 3 Hardware

5.1 FPGA Direct Programming

The first test completed after the power regulation systems were verified was programming of the FPGA with a simple HDL system design. This test's goal was to show that the FPGA was operating properly by the flashing of an on-PCB LED. The FPGA architecture for this test is shown in Figure 5.2.

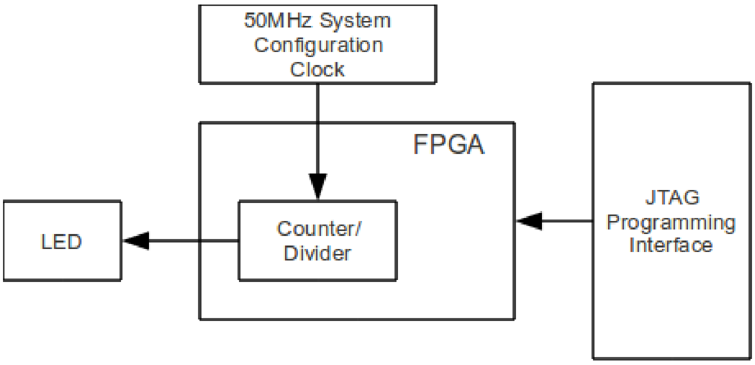


Figure 5.2: FPGA LED Blink Test Block Diagram

This simple FPGA system was created in VHDL using Xilinx ISE software. FPGA I/O pins not used in this design were set to safe default values. An FPGA bit configuration file was produced and programmed directly to the FPGA through the JTAG interface.

This test successfully showed not only that the FPGA could be programmed but also that it was performing a basic task programed in VHDL code. The success of this test was the first step in verifying the other subsystems and devices since the FPGA served as the platform for these further tests.

5.2 FPGA PROM

The FPGA PROM, used to configure the FPGA at power-up of the device, has failed to show proper functionality during testing. Direct programming of the FPGA as described above, however, is possible. Figure 5.3 shows the JTAG and configuration chain of the FPGA and PROM. The JTAG data signal is sent out from the JTAG programmer and passes though the FPGA and the PROM in a serial configuration before returning to the programmer. During device power-on, the PROM loads the stored FPGA configuration onto the FPGA via a separate configuration bus. The fact that the FPGA can be programmed while the PROM cannot is a troubling issue because both devices share the same serial JTAG chain.

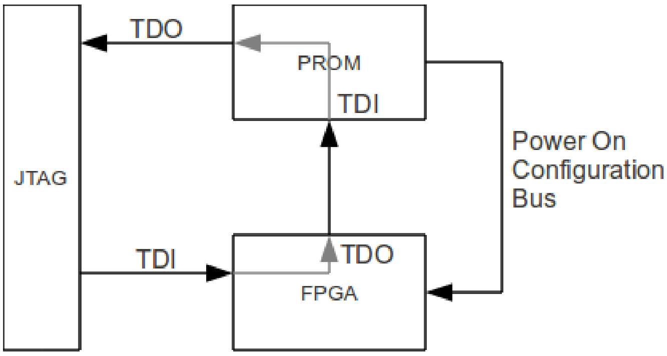


Figure 5.3: Revision 3 FPGA and PROM JTAG Chain

With communication to the FPGA over JTAG verified through programing of the FPGA, it is obvious that the JTAG chain is functioning properly. Additionally, basic functionally of the PROM is present. Simple tasks such as retrieving the device identification number and status registers return the expected results. However, when attempts are made to erase or program the device, the programming software reports either time-out condition or failed verification of the attempted operation.

To investigate this issue, the software settings for programming the PROM in this configuration were verified several times with Xilinx documentation. Following that, the schematics were double checked against a Xilinx-provided reference design that included these components, and no discrepancies were found. Finally, the PCB layout was checked to verify that no routing or shorting issues existed. With all the attempts to solve this problem yielding no results, the PROM has been listed as a non-functioning system for further review. After other subsystems are verified, it will be necessary to revisit this issue. Whether or not the issue is solved in this revision of hardware, it is important that the issue is at the very least identified to prevent recurrence in future revisions of hardware. Proper functionality of the PROM is not only important as a convenience but also so the hardware is not left powered on in an unconfigured state, because hardware damage could occur due to several control lines defaulting to an unsafe state.

For now it is possible for the FPGA to be programmed manually through the JTAG interface directly after a board is powered on. The FPGA configuration is lost when powered down, and the JTAG programming must be repeated every power cycle. This is an adequate solution for conducting testing until the PROM issue is resolved. However, for both streamlined operation and the dangers of hardware damage mentioned above, this issue should be resolved.

5.3 Clocking System

The clocking system was the next to be verified after the FPGA. The high frequency clocks produced by the phased-locked loops (PLLs) and voltage controlled oscillators (VCOs) are necessary in all ADC and DAC testing. To verify their operation, a PLL configuration block was created in the FPGA that programed the various setting registers of the PLLs to produce the system clocks of 880MHz and 750MHz. Each of the two clock systems consists of a PLL connected to a VCO. Each PLL is programmed to control the VCOs with a feedback loop, thus producing the desired clock output. The clocks are then fed to a system clock distribution chip before going to the various devices in the system.

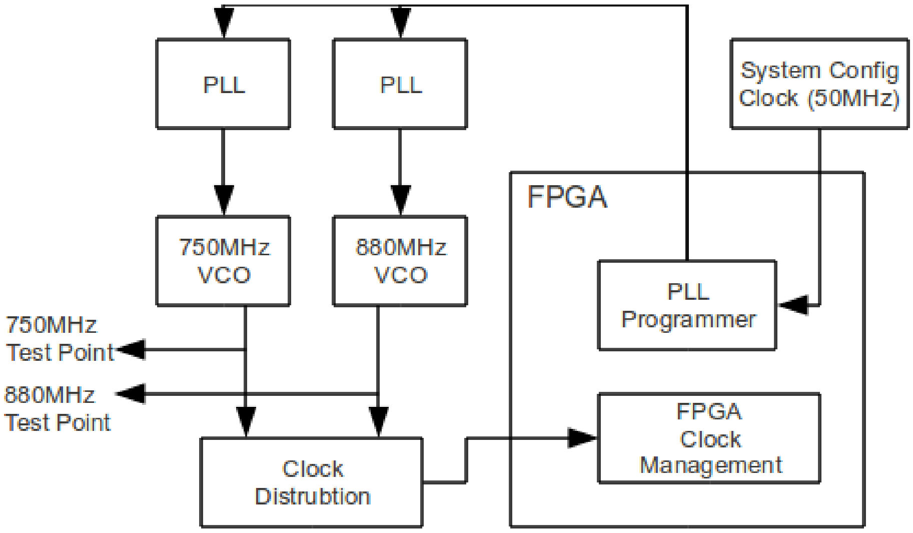
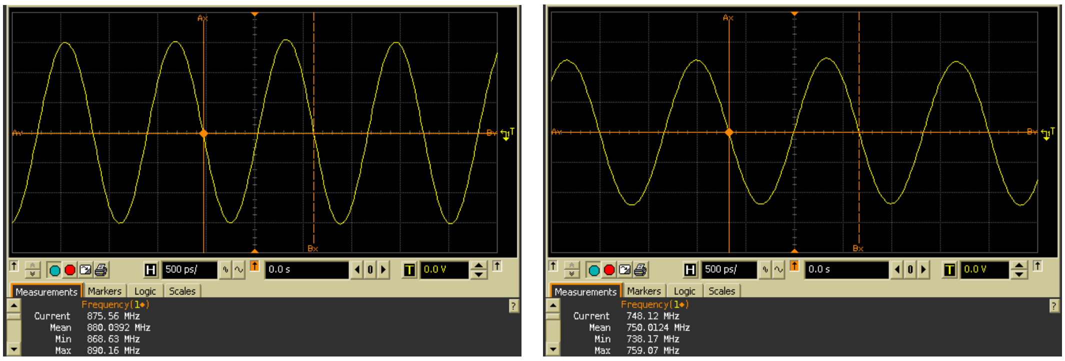


Figure 5.4: Clock Testing Block Diagram

Figure 5.4 shows the system architecture for the clock testing. When the system is first powered on the PLLs are in an unconfigured state and the VOCs do not produce the needed clocks. A crystal oscillator outputting 50MHz is used to drive a block implemented within the FPGA for programming the PLL settings registers. After programming the PLLs, the block waits for a locked status before outputting a clock ready signal to the other blocks in the FPGA system.

Oscilloscope screen captures for each of these clocks can be seen in Figure 5.5. These measurements were taken directly at the output of the VCO before entering the clock distribution chip. Both clocks are accurately produced, indicating proper functionality of the PLLs and VCOs.



```
Frequency(1•)
Current   875.56 MHz
   Mean   880.0392 MHz
    Min   868.63 MHz
    Max   890.16 MHz
```

```
Frequency(1•)
Current   748.12 MHz
   Mean   750.0124 MHz
    Min   738.17 MHz
    Max   759.07 MHz
```

(a) 880MHz

(b) 730MHz

Figure 5.5: System Clocks Oscilloscope Screen Captures

5.4 Transmit Channel and Digital to Analog Converter

After basic system functionality was verified, the next key piece to test was the RF transmission chain of the TART locator. Subsection 4.4 described the digital architecture for interfacing the FPGA with the DAC as well as the theory of operation. To verify the new hardware as well as the theory of operation documented in this thesis, a basic test implementing those methods was conducted.

A basic FPGA system was implemented to initialize the hardware and drive the DAC with an I/Q waveform. A block diagram for this system is shown in Figure 5.6. There are three major components in this FPGA architecture: the initial configuration (blue), waveform storage (green), and double data rate (DDR) DAC interface (red).

The initial configuration system consists of the clock configuration design described in the pre-

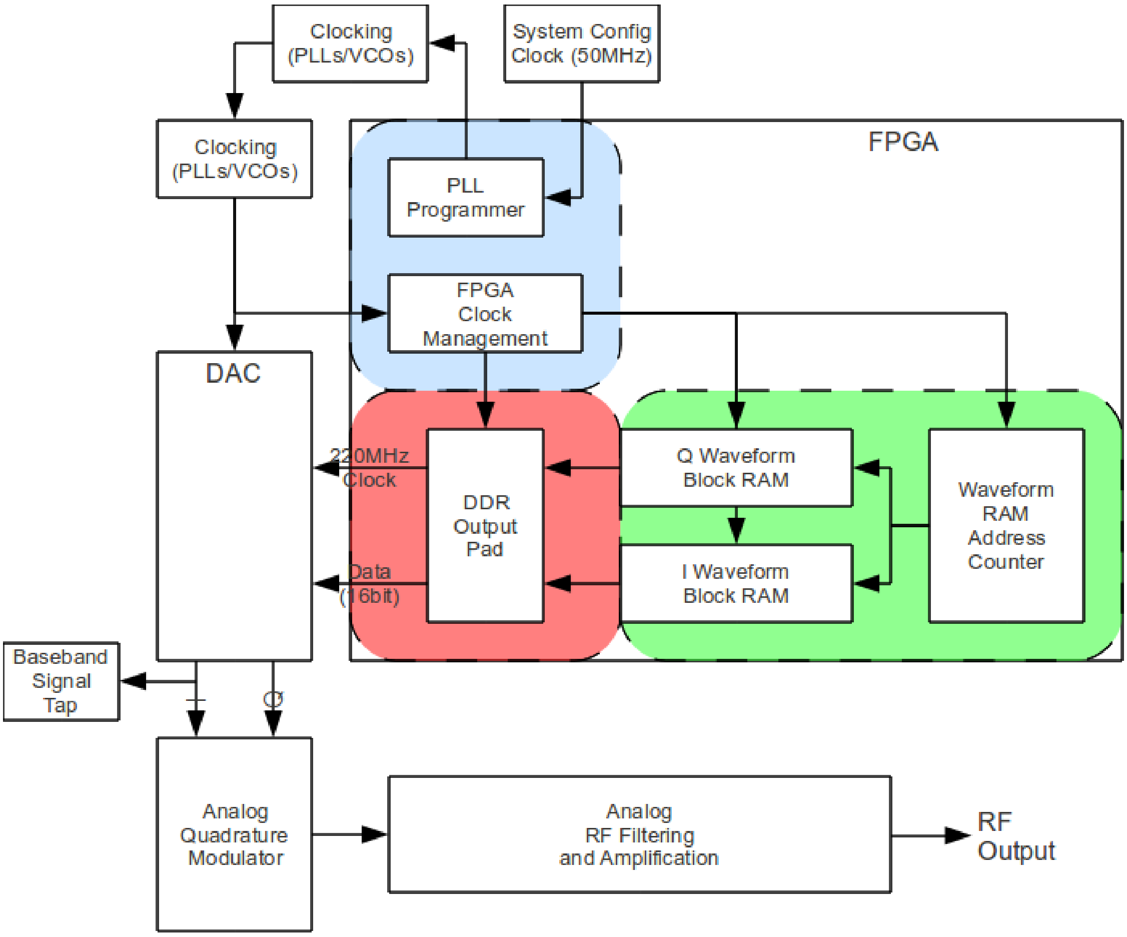


Figure 5.6: DAC Testing Block Diagram

vious section. The PLLs are configured at power-on to produce the required system clocks. The 880MHz clock is run through a divider in the clock distribution chip that outputs a 400MHz clock. This 440MHz clock is used in the FPGA for clocking system blocks relating to the high-speed data interfaces.

The waveform storage holds the digital I and Q waveforms that are output to the DAC. The storage is implemented in the form of FPGA block RAM. The digital samples for the particular waveform to be transmitted are set in the block RAM by reconfiguring the Xilinx block RAM IP-Core using the ISE software at design time, rather then using a loading process at runtime. Although this necessitates an FPGA system rebuild and reprogramming to change the transmitted waveform, it greatly simplifies the system architecture for the purposes of testing. The I and Q waveforms for the desired RF output are generated in MATLAB and set as the initial content for the two block RAMs. A cyclical counter, counting from 0 to 1023, is used to drive the address lines of the two block RAMs for continuous waveform output to the DAC.

The final component implemented in the FPGA for this test design is the DDR DAC interface. This component consists of instantiations of Xilinx's IO Pad resources. This block takes in the single data rate (SDR) I and Q waveforms at 220MHz and outputs to the DAC a double data rate (DDR) signal consisting of interleaved I and Q samples at 440MSPS, shown in the timing diagram in Figure 5.7.

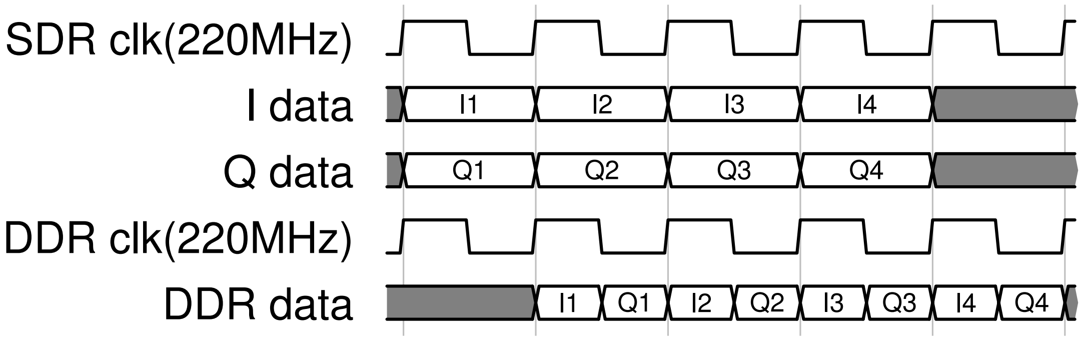
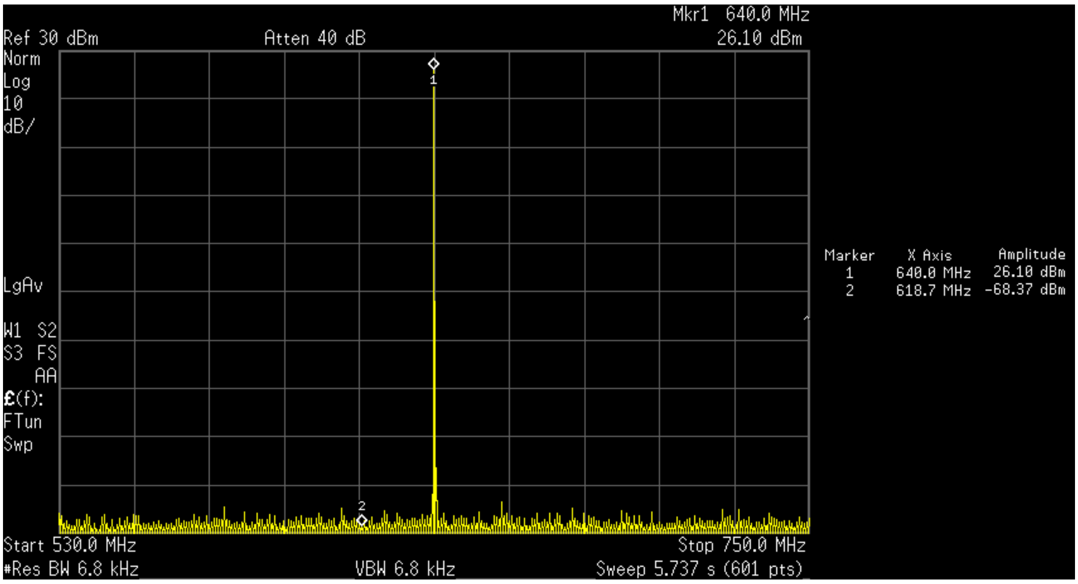


Figure 5.7: ADC FPGA Interface – Digital Timing Diagram

Two tests were conducted with this architecture to verify proper function of the DAC and related circuitry. First, a single carrier signal was generated and output. The I/Q waveform information represented a 640MHz carrier and utilizes 100% of the DAC's dynamic range. The resulting RF output was viewed and recorded with a spectrum analyzer (Figure 5.8).



Marker	X Axis	Amplitude
1	640.0 MHz	26.10 dBm
2	618.7 MHz	-68.37 dBm

Figure 5.8: Revision 3 Single Carrier Output

The final output was found to lie at the desired frequency of 640Hz. Not only was the output signal checked for proper carrier location, power, noise floor, and out-of-band noise and leakage were all observed. The noise floor of the signal fell below the range capable of measurement by the spectrum analyzer in this configuration (approximately –68dBm). Additionally, the LO carrier at 750MHz also fell below the capabilities of the spectrum analyzer. With all these values found to be within the expected ranges for a preliminary test and the output frequency verified for a single carrier, testing using more complex signals could begin.

In addition to the single carrier test, a real-world PPL multicarrier waveform was also tested. The multicarrier waveform was generated in MATLAB and the phases were optimized for a low crest factor[22]. This multicarrier signal ranged from 550–679MHz, except for the stop band from 608–614MHz, and contained 109 individual subcarriers. A floating point double precision MATLAB log amplitude freqency plot of this signal can be seen in Figure 5.9. The 0dB reference for this plot is the maximum dynamic range of the signal. This multicarrier waveform data was used as the basis for producing all digitally quantized DAC and ADC waveforms used in the following tests.

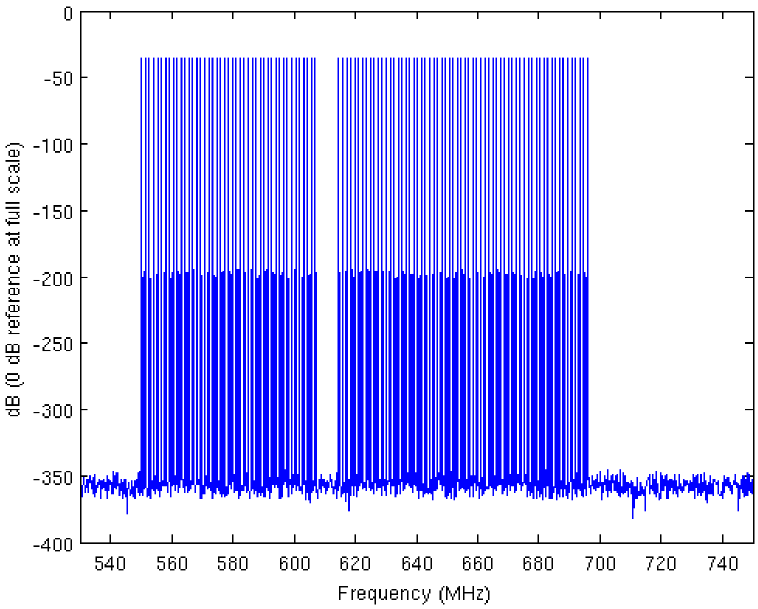


Figure 5.9: MATLAB Generated Multicarrier Waveform

The RF output of the multicarrier signal was observed from 530–750MHz (Figure 5.10). To find the expected amplitude of the subcarriers, the full dynamic range, single carrier test can be used as a basis. By subtracting all factors that would result in subcarrier amplitude loss from the single carrier's amplitude, the expected subcarrier amplitude can be found. The first factor that would result in power loss is the increased number of carriers.

$$\log_2(n_{\text{carriers}}) * 3dB = loss\ dB$$
$$\log_2(109) * 3dB = 20.3dB \tag{5.1}$$

The use of 50% of the DAC's dynamic range also results in a loss.

$$20 * \log_{10}\left(\frac{1}{\text{DynamicRange}}\right) = loss\ dB$$
$$20 * \log_{10}\left(\frac{1}{0.50}\right) = 6.0206dB \tag{5.2}$$

Finally, the increased crest factor must be accounted for. The crest factor in the single carrier case is the ideal 3dB. In the multicarrier case, the crest factor was calculated to be 5.85dB.

$$\begin{aligned} 5.85dB - 3dB &= loss\ dB \\ 5.85dB - 3dB &= 2.85dB \end{aligned} \tag{5.3}$$

From these three factors, the expected subcarrier amplitude can be calculated by subtracting the losses from the amplitude of the single carrier case.

$$\begin{aligned} (\text{single}_{power}) - (\text{carrier\_count}_{loss}) - (\text{dynamic\_range}_{loss}) - (\text{crest}_{lost}) &= \text{subcarrier } dBm \\ 26.1 - 20.3 - 6.0206 - 2.85 &= -3.1dBm \end{aligned} \tag{5.4}$$

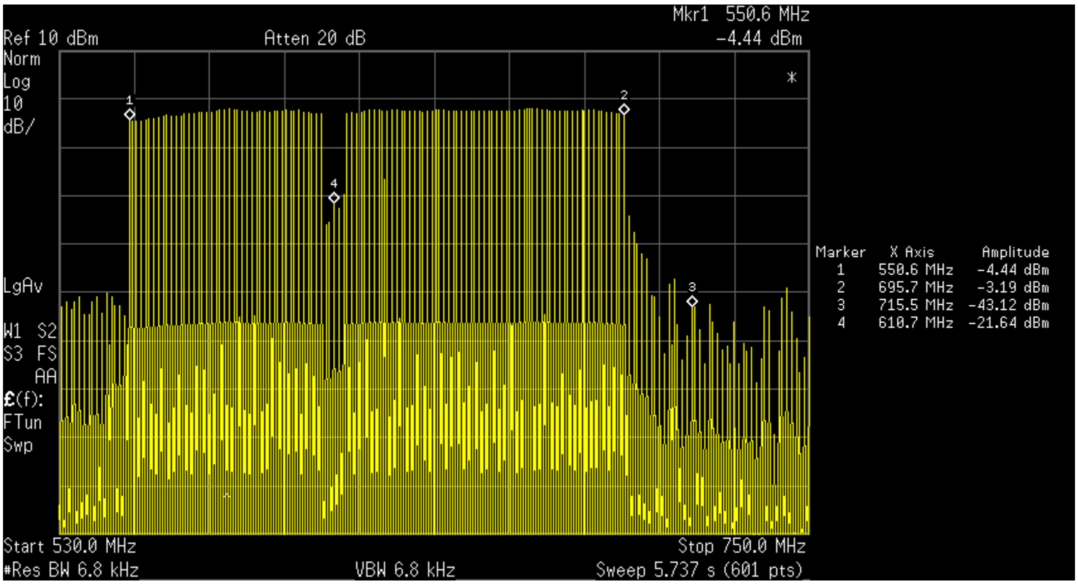
All the measured subcarriers in Figure 5.10 are within 1.5dB of the expected value, with most of the carriers falling much closer. The slightly lower measured carrier level, -4.44dBm, was measured near the beginning of the band pass filter roll-off. Because of this, a deviation of approximately 1.5dB was not unexpected.

The IP3 distortion in the stop band from 608–614MHz was measured at approximately –21.6dBm. This value falls short of the FCC-required –49dBm and will need further investigation to reduce. Increased use of the DAC's dynamic range to achieve a higher subcarrier power will also exacerbate the IP3 issue. Modifications to the PPL multicarrier signal, changes in the amplification chain, and digital preprocessing of the transmitted signal are possible methods of addressing this IP3 issue.

5.5 Receive Channel and Analog to Digital Converter

With the testing of the transmission channel complete, the next step was to test another component of the transceiver system: the receive channel. The testing method for the ADC was similar in many ways to that of the DAC.

Figure 5.11 shows the block diagram for the FPGA system implemented to test the ADC and analog receive chain. Much of the design is similar to that of the DAC system with many functional blocks reused. This design, however, also uses a Microblaze soft-core microprocessor system implemented on the FPGA to read the ADC samples and transmit them to a computer via an RS232 interface. Other than the inclusion of the Microblaze (purple), this design shares the same three major components as the DAC design: the initial configuration (blue), sample storage (green), and the high-speed digital interface (red).



Marker	X Axis	Amplitude
1	550.6 MHz	-4.44 dBm
2	695.7 MHz	-3.19 dBm
3	715.5 MHz	-43.12 dBm
4	610.7 MHz	-21.64 dBm

Figure 5.10: Revision 3 Multicarrier Output

The high-speed digital interface is responsible for retrieving samples from the ADC and conditioning them such that they can be used in the FPGA. Initially, the 14 DDR LVDS signals from the ADC are terminated and changed to single data rate (SDR) using instantiations of Xilinx's IO Pad resources. Following this, a serial to parallel converter is used to lower the data rate of the incoming samples by increasing the bus width to four samples wide. The now conditioned data consists of a four sample (56-bit) wide data bus operating at 110MHz SDR and is ready for storage in RAM.

The waveform storage for this implementation is similar to that in the DAC test architecture. FPGA block ram was again used for sample storage, but now there is only a single block RAM with a 56-bit wide data bus. The on-PCB DIP switches are used to control the write enable for this block RAM. This allows for selective filling of the block RAM when a signal of interest is applied to the RF input.

Finally, the Microblaze system was used to retrieve the ADC samples and output them. The Microblaze was attached to the secondary port of the ADC sample block ram. A continuous code loop cyclically reads the ADC samples. These samples are then output over an RS232 interface where they can be read into a computer for analysis.

The first ADC capture was conducted with the RF input terminated with a 50Ω dummy load.

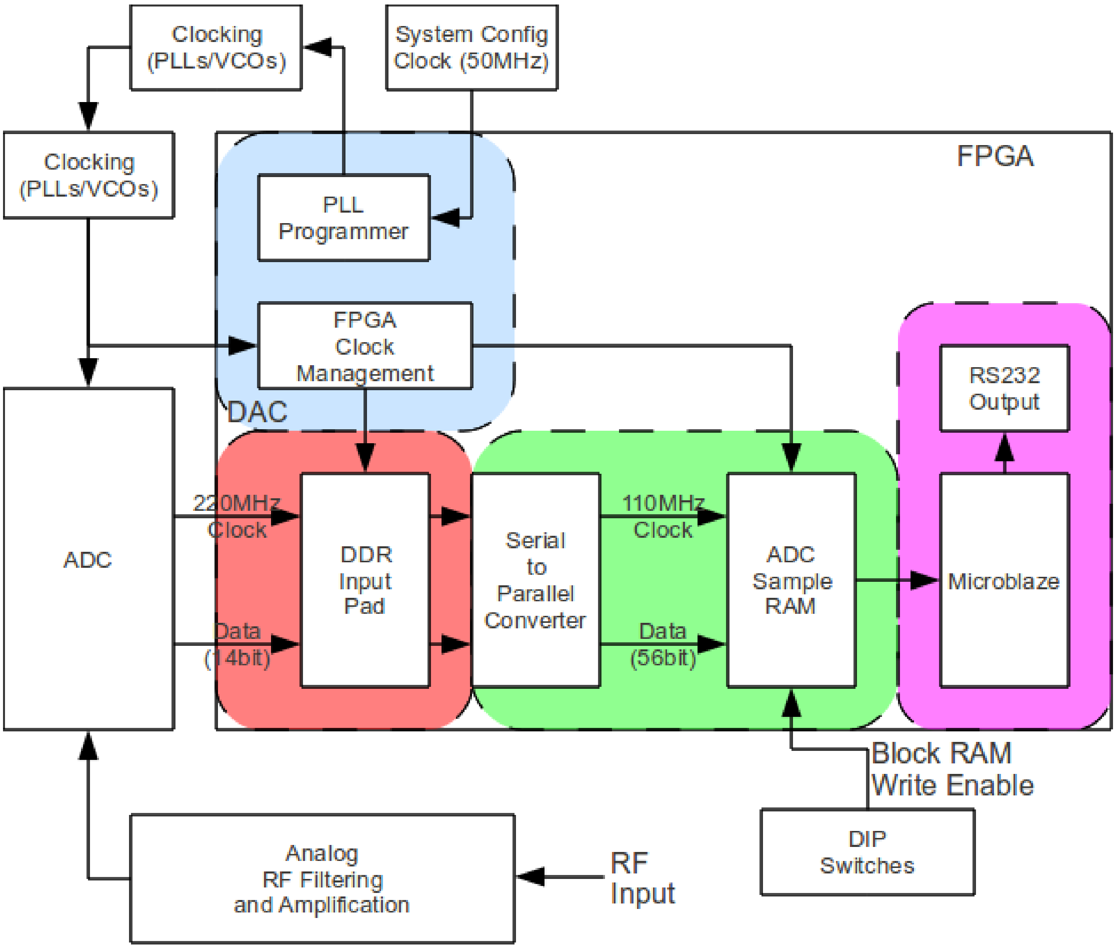


Figure 5.11: ADC Testing Block Diagram

The results of this capture showed the noise and frequency response characteristics of the receive channel and ADC when no input signal is applied. The time domain ADC samples were read into MATLAB, where analysis of the data could be completed.

Figure 5.12 shows the captured noise as the blue line from 530–750MHz. The amplitude scale for this frequency plot, as well as all of the following ADC plots, is a log power with 0dB reference being the maximum dynamic range of the ADC's digital data. From this plot, it can be shown that the ADC is capable of approximately 70dB of SNR.

A sixth order polynomial curve was fit to the noise data, shown as the red line in Figure 5.12, to help illustrate the frequency response of the channel. This curve clearly showsthat the frequency response is not flat. However, this issue can be addressed through further modifications of the receive channel's analog circuitry and post-processing of captured signals with calibration data.

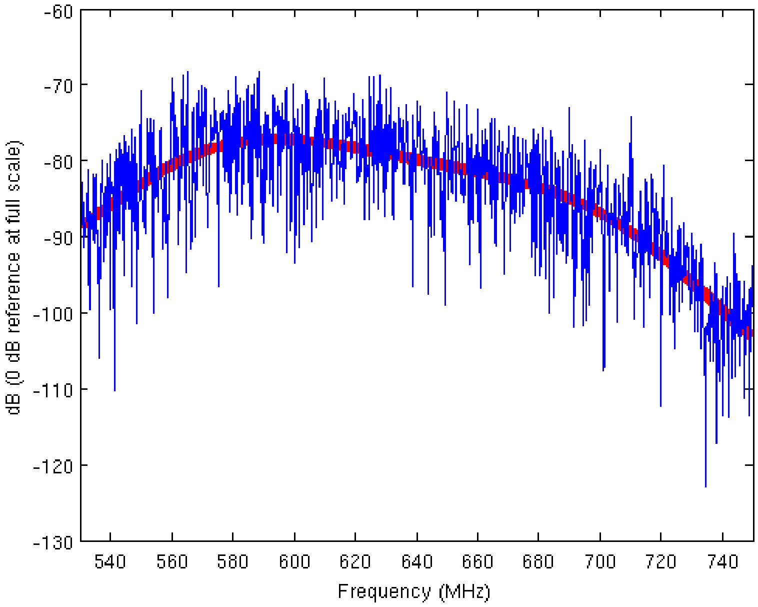


Figure 5.12: 50Ω Terminated ADC Noise Capture

Next, a single carrier input was applied to the RF input. Due to the wired nature of this test, care was taken to select a power level that simulated received signals from an antenna. A function generator was used to create a –50dBm single carrier at 700MHz. Again, the ADC samples were read into MATLAB. The signal was found to use approximately 33% of the ADC's dynamic range.

Figure 5.13 shows a log amplitude frequency plot of the ADC sample data. Rather than showing the flipped baseband image as explained in Subsection 4.3.1, the plot was flipped and frequency shifted to represent the RF input signal for easier visual interpretation. The ADC data shows a strong peak at 700MHz with a SNR of approximately 55dB. Given that only 33% of the ADC's dynamic range was utilized, an SNR of 55dB is an appropriate value. The roll-off in power on either side of the 700MHz peak is due to the sinusoid not falling on an integer multiple of the FFT bins.

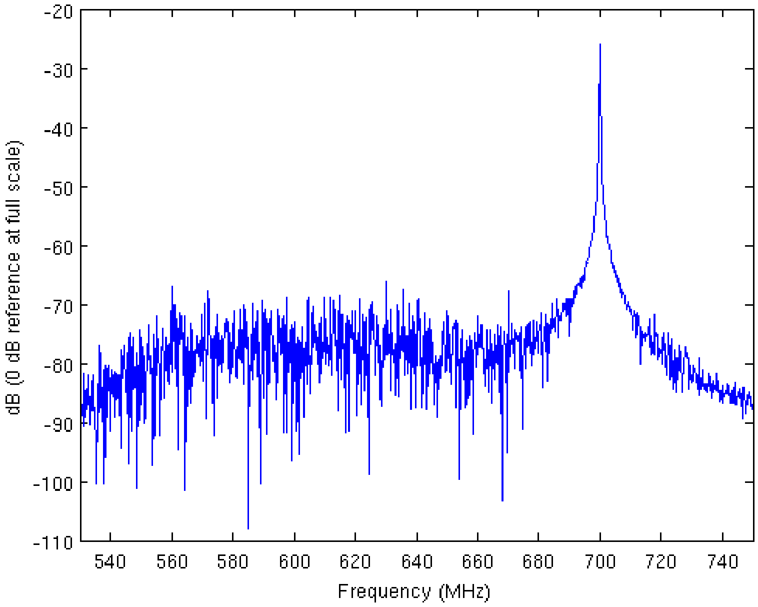


Figure 5.13: ADC Single Carrier Captured Data Plotted in the RF Band

For the final ADC test, a PPL multicarrier signal was used as RF input to the system. The multicarrier waveform for this test was the same as that shown in Figure 5.10. To simulate the attenuation of an over-the-air RF transmission and to prevent saturation of the ADC's dynamic range, 72dB of attenuation was applied before the signal was input to the system.

The signal was again captured and processed in MATLAB. Figure 5.14 shows the log amplitude frequency plot of the ADC data. Again, this data was flipped and relabeled to represent the RF input signal. The signal has approximately 40dB of SNR in this case. The lower SNR is to be expected when compared with the single carrier due to the higher crest factor. Additionally, the non-flat frequency response of the receive channel is exposed by the multicarrier signal.

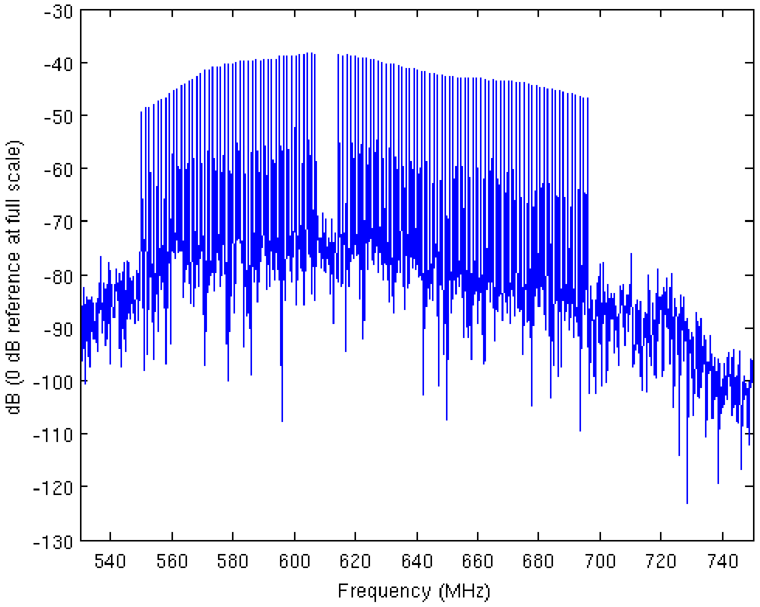


Figure 5.14: ADC Multicarrier Captured Data Plotted in the RF Band

By applying the sixth order polynomial derived from the noise data in Figure 5.12, a crude but effective method of calibrating the frequency response of the receive channel and ADC is achieved. The frequency location with minimum SNR was retained, and the reaming had their amplitude raised according to the calibration curve. Figure 5.15 shows the multicarrier ADC capture with this calibration data applied. Even with this crude calibration, a noticeable improvement of approximately 6dB less ripple in frequency response is obtained.

5.6 Dynamic Waveform Generation Block

The final major subsystem relating to the transceiver system in the revision 3 locator is the dynamic waveform generation block. To test this block, several I/Q waveforms were generated and transmitted through the DAC and amplification chain. The full system architecture with the addition of the dynamic waveform generation block is shown in Figure 5.16. This architecture consists of the ADC and DAC test architectures, with the addition of the dynamic waveform generation block. The dynamic waveform generation block is connected with the ADC sample RAM as an input and the

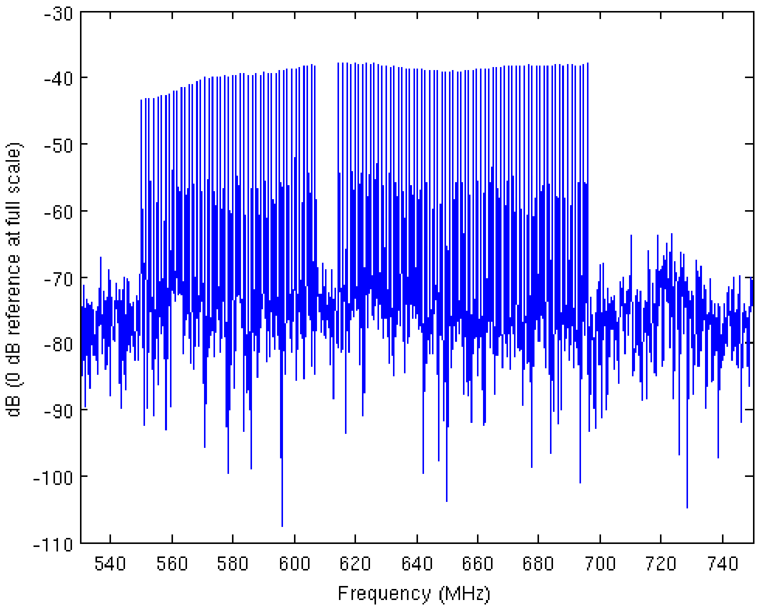


Figure 5.15: ADC Multicarrier Captured Data – Basic Channel Calibration Applied

DAC I and Q waveform RAM as the output. The dynamic waveform generation block is enabled though the use of one of the on-PCB DIP switches.

To accurately assess the proper operation of the dynamic waveform generation, synthetically produced ADC sample data was used initially. This synthetic data removed any imperfections the receive channel may have introduced. The synthetic data was produced from the floating point double precision data shown in Figure 5.9 by digitally quantizing to the 14-bit dynamic range of the ADC. The synthetic ADC data is shown as a spectral plot in Figure 5.17.

This data was preloaded at design time into the system's ADC sample RAM and run through the generation block. The output was then transmitted and observed from 530–750MHz (figure 5.18). It is obvious from the subcarrier's amplitude of approximately –18dBm that the full DAC dynamic range was not utilized. This is due to the conservative static scaling schedule used in the dynamic waveform generation block and will be addressed through a future dynamic scaling schedule. However, the output is a faithful representation of the input multicarrier waveform and shows low distortion and no obvious artifacts.

The second round of tests was conducted with the data acquired during testing of the ADC. The

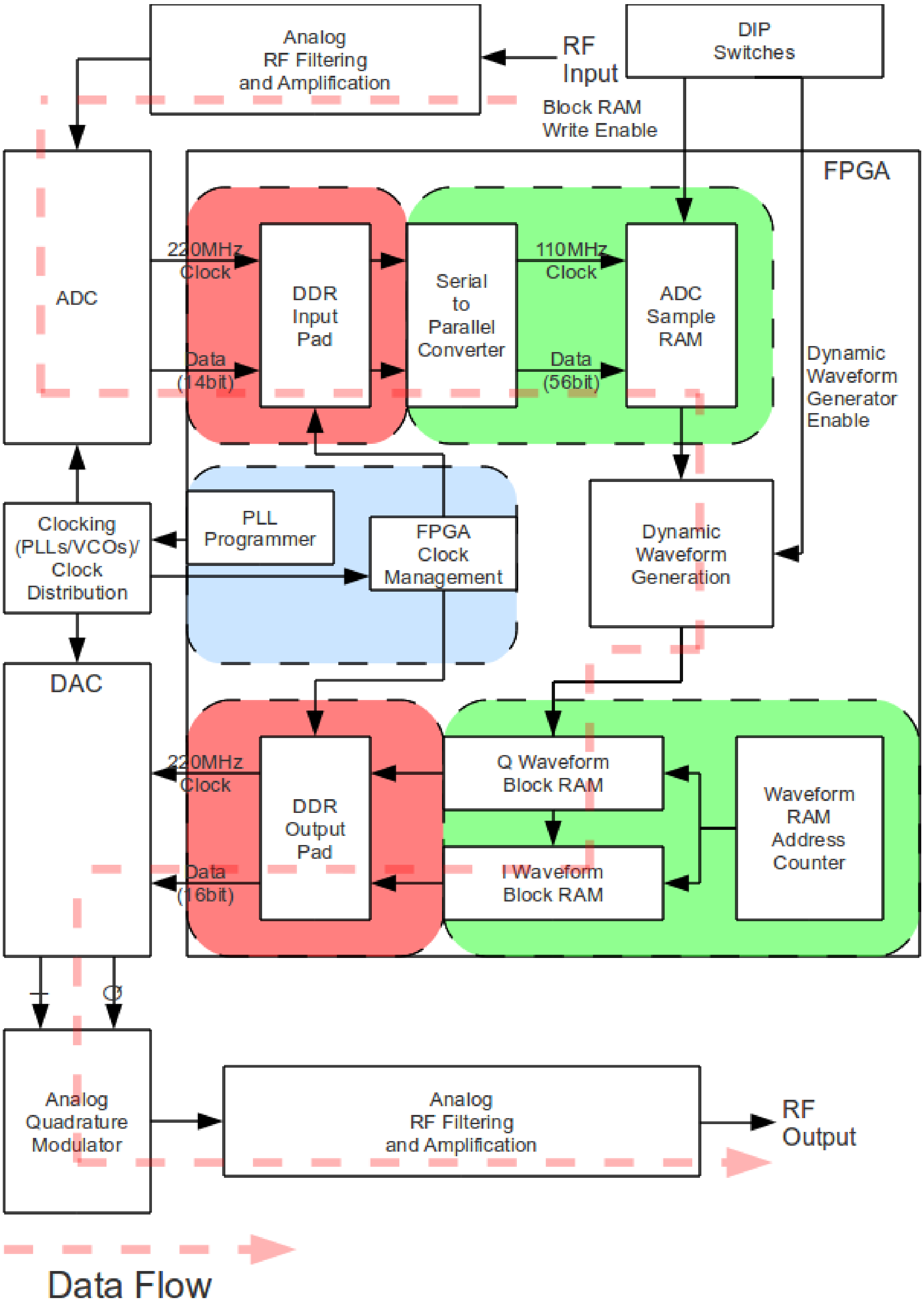


Figure 5.16: Dynamic Waveform Generation Testing Block Diagram

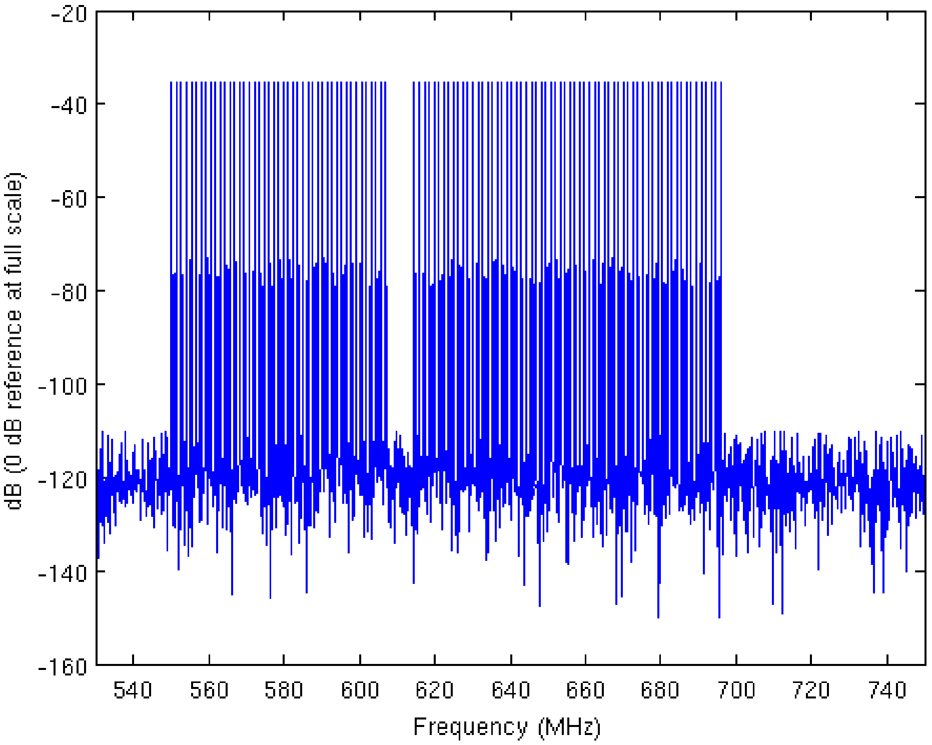
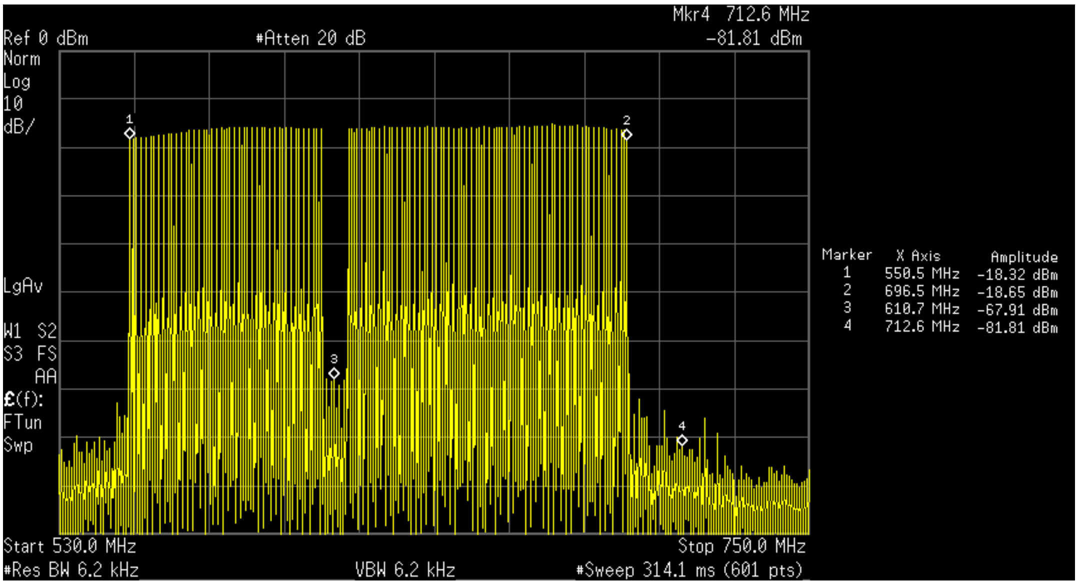


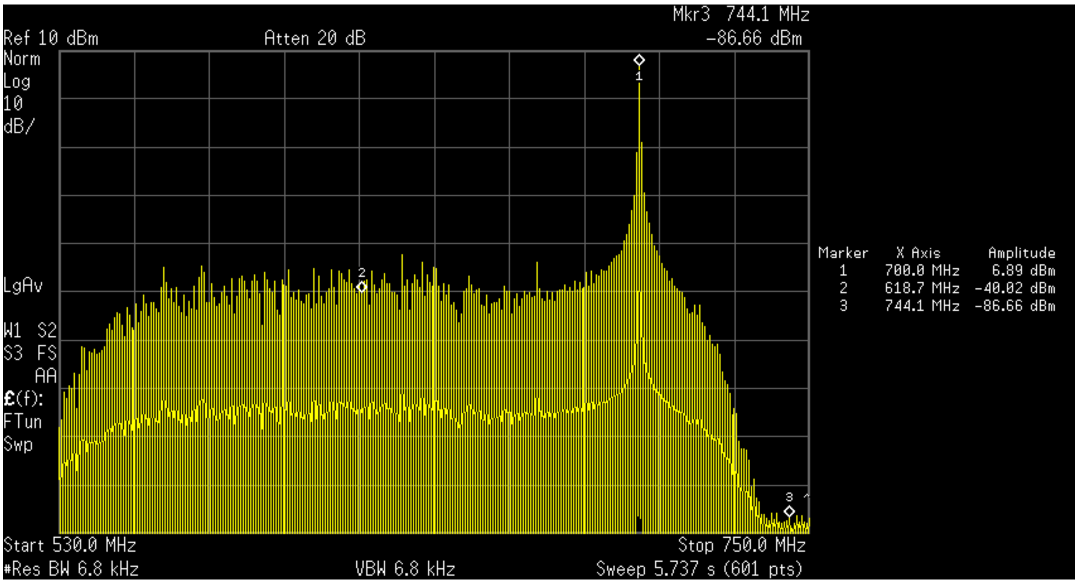
Figure 5.17: Synthetic ADC Data for Input to Dynamic Waveform Generation Block



Marker	X Axis	Amplitude
1	550.5 MHz	-18.32 dBm
2	696.5 MHz	-18.65 dBm
3	610.7 MHz	-67.91 dBm
4	712.6 MHz	-81.81 dBm

Figure 5.18: Output of Dynamically Created Waveform from Synthetic Data

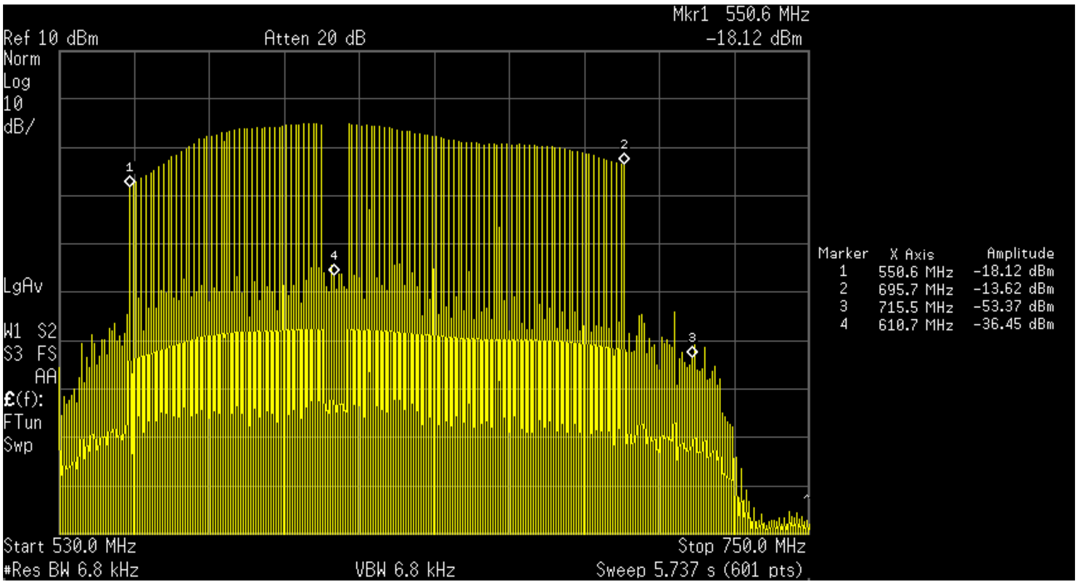
first signal was the –50dBm single carrier located at 700MHz (Figure 5.13). The transmission of the I/Q waveform generated from the ADC data is shown in Figure 5.19. The single carrier was accurately reproduced at 700MHz. However, the noise floor was significantly higher in this test compared with the synthetic test. This effect will be addressed by several features that have yet to be implemented in the dynamic waveform generation block. First, frequency masking provides the ability to remove all signals other then ones located at frequencies of interest (700MHz in this case). Second, a dynamic scaling schedule will allow full use of the DAC's dynamic range.



Marker	X Axis	Amplitude
1	700.0 MHz	6.89 dBm
2	618.7 MHz	-40.02 dBm
3	744.1 MHz	-86.66 dBm

Figure 5.19: Output of Dynamically Created Single Carrier Waveform

For the final dynamic waveform generation test, an I/Q waveform was generated from the data captured during the multicarrier ADC test (Figure 5.14). The output of this test is shown in Figure 5.20. Again as in the single carrier case, the noise floor is significantly higher compared with the synthetic case. However, the correct multicarrier waveform was produced. The frequency response of the ADC channel can be seen in the ripple of the multicarrier signal. Future improvements to the dynamic waveform generation will include methods of flattening frequency response through either receive channel calibration or frequency carrier normalization.



Mkr1 550.6 MHz
-18.12 dBm

Ref 10 dBm
Atten 20 dB
Norm
Log
10
dB/
LgAv
W1 S2
S3 FS
AA
£(f):
FTun
Swp

Marker	X Axis	Amplitude
1	550.6 MHz	-18.12 dBm
2	695.7 MHz	-13.62 dBm
3	715.5 MHz	-53.37 dBm
4	610.7 MHz	-36.45 dBm

Start 530.0 MHz
Stop 750.0 MHz
#Res BW 6.8 kHz
VBW 6.8 kHz
Sweep 5.737 s (601 pts)

Figure 5.20: Output of Dynamically Created Multicarrier Waveform

5.7 Over the Air Transmit and Receive

A final test was conducted on the revision 3 hardware that encompassed all the transceiver subsystems: the transmit channel, the receive channel, and the dynamic waveform generation block. A multicarrier waveform generated from synthetic data by the dynamic waveform generation block was transmitted over the air from revision 3 locator hardware, and then received by a second locator.

This test was conducted in one of the PPL laboratories at WPI as shown in Figure 5.21. Each unit was connected to a commonly used wide-band antenna of the PPL project referred to as the ”bow tie”. This antenna, which is often used in PPL testing for its horizontally uniform antenna pattern, provides 3dBi of gain. The antennas were placed approximately 3m apart and 1.5m from the ground.

The first data capture was conducted without the presence of a PPL multicarrier transmission to assess the state of the 530-750MHz RF band. The output of this capture is shown in Figure 5.22. There was a presence of background noise throughout the band compared with the terminated blank capture (Figure 5.12). Additionally, there were two distinct transmissions at approximately 570MHz and 720MHz. Each transmission is the signal from a digital television broadcast.

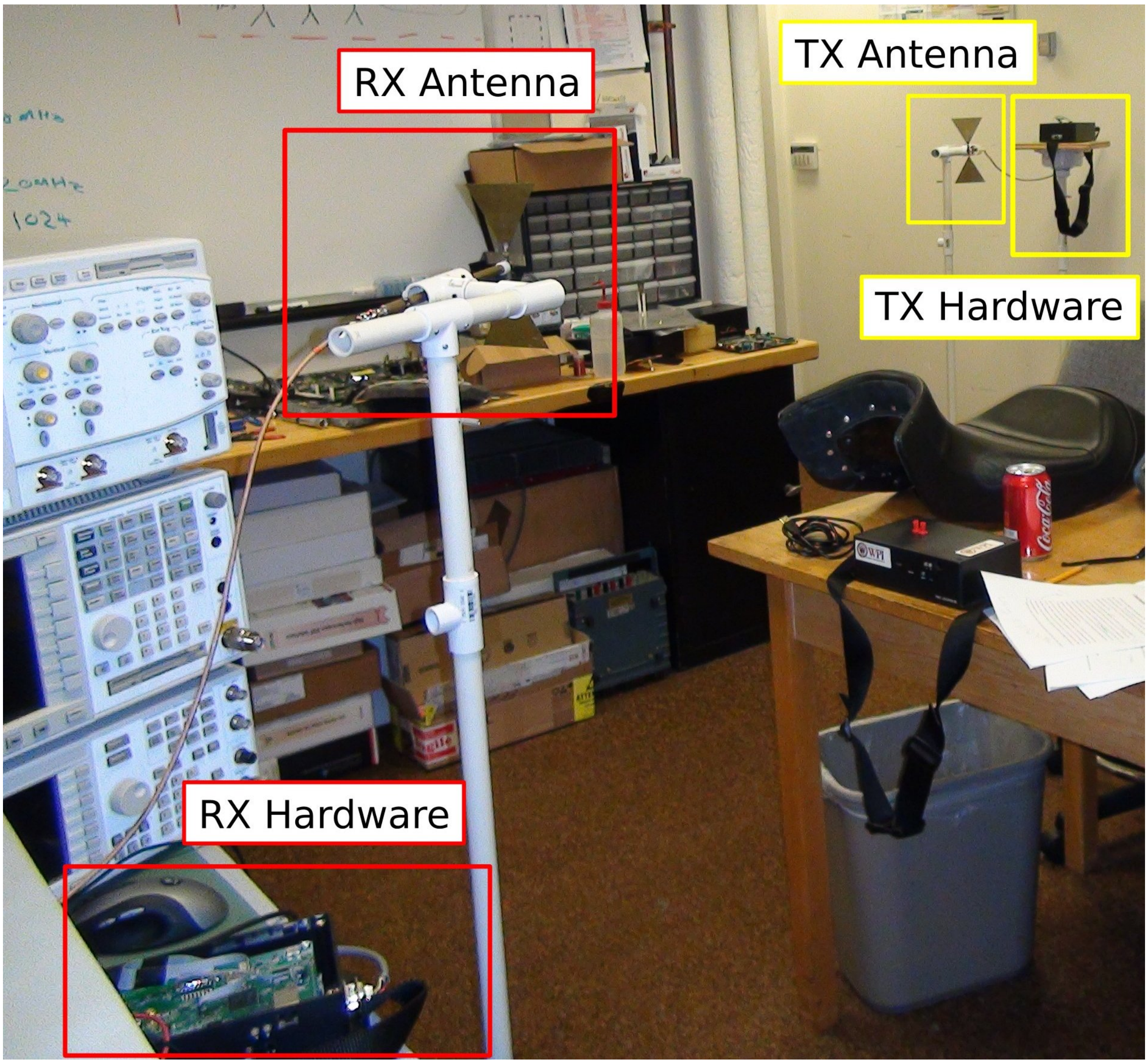


Figure 5.21: Over the Air Test Setup

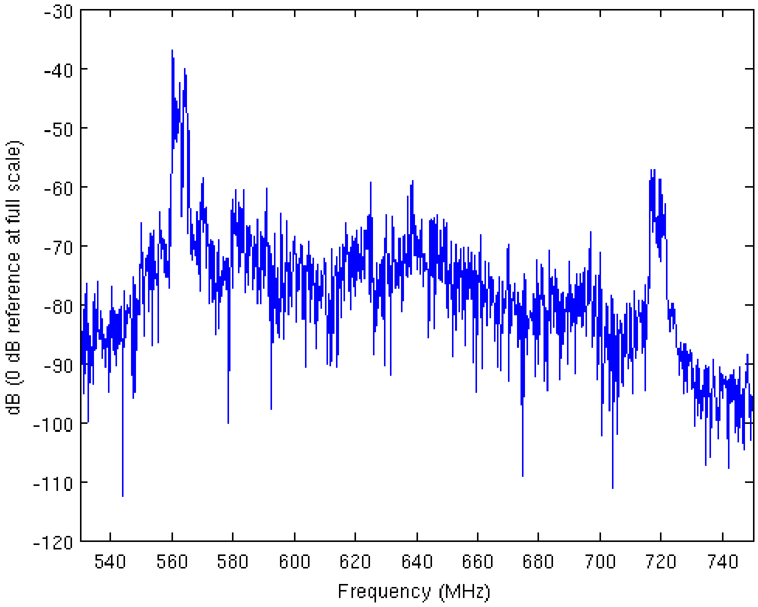


Figure 5.22: Noise Capture of PPL Spectrum

With knowledge of the RF band, a capture was conducted with a PPL multicarrier transmission. The signal used for transmission was the multicarrier dynamically created from synthetic ADC data (Figure 5.18). This signal was chosen to illustrate the effectiveness of the dynamic waveform generation block and also because it complies with FCC out-of-band emission requirements. To avoid saturation of the receive front end and ADC due to the close proximity of the antennas to each other, 28dB of attenuation was applied to the transmitted signal. The captured results are shown in Figure 5.23. Although there is significant amplitude fluctuation, the PPL multicarrier signal is clearly visible above the noise floor.

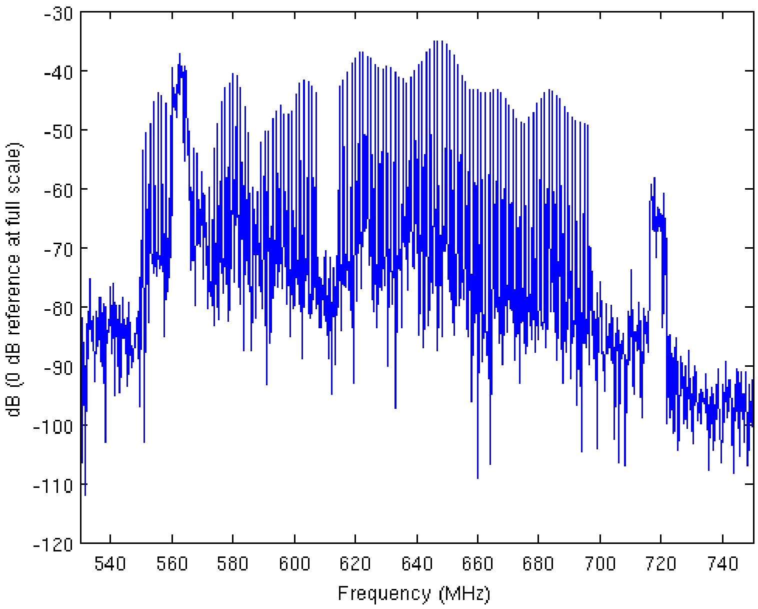


Figure 5.23: PPL Multicarrier Over the Air Transmission – Received ADC Data

5.8 Current System Status

Currently not all the subsystems of the new hardware have been fully tested. There is still work to be done on several of the subsystems that have been tested in addition to continuing full subsystem verification. Table 5.1 shows the various subsystems and their current status of operation and testing.

System	**Status**
PROM	Not functioning
FPGA	Verified
Power Regulation	Verified
Battery Charging	Verified
Clocks (VCOs/PLLs)	Verified
Transmit Chain	Verified
Receive Chain	Verified
MicroSD Card	Verified
RS232	Verified
LEDs	Verified
GPIO	Verified
DIP Switches	Verified
Pressure Sensor	Verified
Ethernet	Untested
SRAM	Untested
Xemics Wireless	Untested
CEL Wireless	Untested
WPI SRM	Untested
On-board Accelerometer	Untested
nIMU Interface	Untested
NavChip	Untested

Table 5.1: Current System Status

Chapter 6

Conclusion

This thesis involved the design, implementation, and testing of a new revision of the PPL hardware system driven mainly by the need for supporting the new TART algorithm. This new hardware is referred to as revision 3 to distinguish it from the previous PPL hardware systems. In addition to TART functionality, improvements on previous revisions were introduced. The design process considered many problems unique to this revision, including the addition of full transceiver capabilities as well as the new unified hardware approach system architecture. In initial testing, the revision 3 hardware has shown itself to be a promising platform for continued indoor location research.

The implementation of the TART location algorithm in a field deployable system was a major goal of the work of this thesis. The TART algorithm promises to yield performance increases over σART. With TART's TOA-based methods, several hurdles had to be overcome relating to data movement throughout the system, clock synchronization issues, as well as countless others. With the new hardware, research focusing on improving the TART location method can continue in a more streamlined manor.

The inclusion of full transceiver capabilities to support the bi-directional TART algorithm in this hardware is a first for the PPL project. Although full transceiver functionality has been present for nearly two years in the PPL transceiver unit, moving this functionality into a mobile platform was no small task. Obstacles ranging from PCB size, power consumption, and interfacing with the Spartan 6 FPGA all had to be overcome before the hardware could come to fruition. Not all the issues relating to the receive and transmit hardware have been fully addressed, but the preliminary testing shows that the design process resulted in a successful hardware implementation.

Another key aspect that the work of this thesis has brought to the PPL project is the concept of the unified hardware approach (UHA). This system configuration will utilize the new revision 3 hardware throughout the PPL system architecture. This revision was designed to satisfy the needs of all the hardware roles, including the mobile locators and reference stations. During system testing, deployment of the hardware will become a simpler and quicker task compared with previous system configurations. In addition, a single hardware platform that supports the entire PPL project's needs will result in less duplicated effort and a more streamlined research and development process.

In addition to the major hardware subsystems that received attention during this design process, various sensors and peripherals included in previous revisions, such as the barometric sensor and digital radio, have been updated and included in this design. Currently, several of these have been tested to show basic functionality, and work will continue to fully verify their operation. Although not all the peripherals have been fully tested, basic communication has been established showing no major hardware flaws.

After development of schematics and PCB layouts, an initial production run of five hardware units was completed. These units were run through preliminary system testing to show functionality and expose any issues in the design. The following section mentions some of the issues encountered during this testing and presents recommendations for resolving these and for additionally system testing.

6.1 Future Work

Although the revision 3 hardware has reached a state such that preliminary TART testing can begin, there are several outstanding issues that still need to be addressed. These issues range from simple testing of components to unresolved issues involving the transceiver channel.

The transceiver channel has been tested to a point where basic functionality has been verified. However, during the course of this testing, several issues were uncovered that will need to be addressed before the new hardware is field test ready.

The receiver channel, including the analog RF front end and ADC, has shown the ability to communicate with the FPGA and send digital samples from the ADC into the fabric of the FPGA.

During the preliminary testing, a non-flat frequency response of the receive channel, containing approximately 10dB of ripple, was exposed during the data captures involving the PPL multicarrier signal. Additionally, although not directly observed, it is a safe assumption that the phase response of the channel is similarly imperfect. There will be two methods of addressing these issues to achieve optimal system performance. First, changes to the passive components in the analog receive chain can help to flatten the frequency and phase response. Second, digital calibration using test signals and ADC data will allow any remaining imperfections to be corrected at the time of processing on a PC.

The transmit channel, including the analog chain, amplification stages, and DAC, has shown excellent performance during testing in terms of power output, signal generation, and channel response. Again, there will need to be in-depth testing to obtain data for calibration to correct any remaining channel imperfections much like with the ADC. One major unresolved issue is the presence of high-amplitude third order intermodulation product distortion (IP3). Due to the nature of the PPL ranging signal (a wideband equally spaced series of subcarriers) IP3 distortion is an unavoidable artifact that is particularly pronounced when operating the power amplifier near saturation. A simple solution to this issue is to reduce the signal power. Because the IP3 distortion artifacts decrease at a rate 3x that of the power level of subcarriers, a drop of approximately 10dB in the multicarrier waveform would bring the out-of-band IP3 artifacts below the FCC required level. A more complex solution that sacrifices less transmission power is digital predistortion of the transmitted signal prior to entering the DAC and transmit chain. Since the PPL system transmits a static signal, active feedback would not be required to achieve usable gains in distortion cancellation, and possibly could be completed on a PC at the time of generation of the signal rather than in hardware.

Now that the majority of system-critical hardware subsystems have been verified, it is recommended that preliminary field testing begin with the new hardware. The most basic first step of testing will include two units using a common clock. By utilizing only two units and a common clock, several more complicated issues relating to the final system architecture are circumvented for this early test. With the use of two units there is less need for complicated software control and interfacing between multiple units, thus easing the implementation of the software and FPGA architectures. Second, the use of a common clock removes the need for any clock drift prediction, simplifying the TART transaction from three transmissions down to only two. Without concerns of clock drift, the TART transaction can be spread out over a theoretically infinite amount of time, provided that the RF environment remains unchanged. This will reduce the need for high speed control synchronization, allowing for simpler software design.

Following this basic TART testing, the next step should be to begin tackling the issues related to a full, field-ready system. Among these issues are the linear clock drift prediction, system-wide timing synchronization, and the implementation of the software stack to support this testing. The software stack involves writing drivers for each of the various subsystems and sensors included in this design. Additionally, a real-time software system complete with a digital communications protocol will need to be written to support the needs of PPL system field testing. In addition to these software elements, additional FPGA functional blocks will need to be implemented to provide future system functionality as well as to provide tight coupling between the software and hardware designs of the FPGA.

One final issue that needs to be addressed is the configuration PROM, which loads the FPGA design at power-on of the hardware. Initial attempts to program the PROM were unsuccessful. Basic communication with the PROM such as retrieval of the devices identification number was possible showing that the part was at least partially functional and the JTAG chain was working. However, every attempt to program or erase the data on the PROM were unsuccessful. Several different methods of programming were tried, but the cause of this issue has not been identified. Until resolved, the FPGA must be programmed manually over the JTAG interface following each power cycle of the hardware. Future efforts to possibly restore functionality, or at the very least identify the cause of this issue, must be carried out.

Although revision 3 has yet to reach the point of maturity at which a complete system field test is possible, it is safe to say that the design process has successfully addressed the goals put forth in this thesis. From the current state of the hardware, it will be possible to show basic TART functionality with minimal additional work, followed by full system field test implementation.

It is unlikely that revision 3 will be the final revision of PPL hardware produced for testing purposes, but it will help provide the platform necessary to continue pushing the boundaries of indoor location research and help to achieve the ultimate goal of placing a life saving system in the hands of first responders.

References

[1] Sandeep Ravindran. Fire Escape. *Popular Science*, March 2010.

[2] Hauke C. Dämpfling. *Design and Implementation of the Precision Personnel Locator Digital Transmitter System*. Master's thesis, Worcester Polytechnic Institute, Dec 2006.

[3] Vincent Amendolare. *Transactional Array Reconciliation Tomography for Precision Indoor Location*. PhD thesis, Worcester Polytechnic Institute, 2010.

[4] AD9122: Dual, 16-Bit, 1200MSPS, TxDAC+ Digital-toAnalog Coverter (Rev 0), 2010.

[5] XILINX Inc. Extended Spartan-3A FPGA Family Product Table, 2008.

[6] XILINX Inc. Spartan-6 FPGA Family Product Table, 2010.

[7] Edward J. Canty. Six firefighters missing in blaze at vacant building. *Worcester Telegram and Gazette*, December 1999.

[8] J. Orr and D. Cyganski. *Firefighter and other Emergency Personnel Tracking and Location Technology for Incident Response*. Technical report, Worcester Polytechnic Institute, July 2001.

[9] Vincent Amendolare. *Synchronization in an Indoor Precision Location System*. Master's thesis, Worcester Polytechnic Institute, 2007.

[10] J Coyne, D Cyganski, and J Duckworth. Fpga-based co-processor for singular value array reconciliation tomography. In *16th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, April 2008.

[11] SEMTECH. XE1205, 180 MHz 1GHz, Low-Power, High Link Budget Integrated UHF Transceiver, 2008.

[12] NANO IMU Product Specification Users Guide (Rev B), 2009.

[13] CEL. ZIC2410 Datasheet (Rev A), 2009.

[14] Texas Instruments. ADS5474 - 14-Bit, 400-MSPS Analog-to-Digital Converter (Rev A), 2008.

[15] XILINX Inc. Spartan-3A DSP FPGA Family Data Sheet (DS610), 2010.

[16] XILINX Inc. Spartan-6 FPGA Data Sheet: DC and Switching Characteristics (DS162), 2010.

[17] V. Amendolare, D. Cyganski, and R. Duckworth. Wpi precision personnel locator system: Inertial navigation supplementation. In *Position Location And Navigation Symposium*, January 2008.

[18] InterSense. InterSense NavChip, 2010.

[19] Analog Devices. ADXL 330: Small, Low Power, 3-Axis 3 g i MEMS Accelerometer, 2007.

[20] N. Bryner D. Madrzykowski D. Stroup and J. Lee. Performance of Personal Alert Safety Systems in Laboratory and Full-Scale Experiments. Technical report, National Institute of Standards and Technology, 2002.

[21] Xilinx. LogiCORE IP Fast Fourier Transform v7.1, 2010.

[22] Stephen Boyd. Multitone signals with low crest factor. *IEEE Transactions on Circuits and Systems*, CAS-33(10):313–319, October 1986.