Project:

S24-41 - FPGA Implementation of a RADAR System

Customer: Dr. Jensen, Sarah

SME: Dr. Walling, Jeffery

Team:

Paul Broome, Allison Pitzl, Tommy Skidmore, Elliott Kroll, Vincent Snell

Date: 2024/04/15

Table of Contents

Problem Analysis

Objective

The project's original goal was to explore Software Defined Radio (SDR) technologies and techniques paired with Synthetic Aperture Radar (SAR) technologies techniques and methods. The end result is a system that can accurately measure the speed of a moving/stationary object up to 2 m/s, as well as differentiate an object from the background with an unambiguous range of 10m and with at most a 3m resolution. This was to be accomplished with the Zynq UltraScale+ RFSoC DFE ZCU670 Evaluation Kit.

Instead of mechanical movement being used to create SAR images, a sweep of the area of interest is performed with a linear antenna array. The array is connected to the RFSoC via its breakout board included in the Evaluation Kit. SARDINE runs the calculations on the RFSoC in near real time to both transmit (TX) using beamforming and receive (RX). The resulting data output was to be a heatmap displayed on a computer.

Requirements

The requirements were as follows:

- Use the Zynq UltraScale+ RFSoC DFE ZCU670.
- Attempt to implement a system that utilizes Synthetic Aperture Radar.
- Develop a radar system to be implemented on the RFSoC.

Constraints

The biggest constraint of the project was the RFSoC that had to be used. The Zynq UltraScale+ RFSoC DFE ZCU670 is a beta board, meaning that support for the board on software applications either doesn’t exist or is not available to the public. This was far and away the largest obstacle in the development of the project. The radar system itself had to be designed in a software environment such as Vivado or MATLAB/SIMULINK, but unfortunately both software programs lacked any support for the board making it almost impossible to efficiently implement the designed software onto the FPGA.

Engineering Standards

Throughout the entire design process, detailed notes were taken on the decisions and designs. All software information (including test runs and development code) was stored on a team GitHub that was regularly updated. Of course, when working with radio federal regulations on frequency usage must be considered. The Radar working frequencies were confined to the ISM (Industrial, Scientific, Medical) bands. ISM (Industrial, Scientific, and Medical) bands refer to radio frequency bands reserved internationally for unlicensed use, primarily for industrial, scientific, and medical purposes.is also a maximum power requirement for non-registered devices, but the project operated far below this power threshold.

Updated Final Detailed Design

Design Documentation

Interfaces:

- Hardware

The Hardware required for this project is as follows in Table 1:

Table 1

	Inputs used	Outputs used	Constraints
Zynq UltraScale+ RFSoC DFE ZCU670 Evaluation Kit	USB 3.0	USB 3.0, SD card	1GHz bandwidth (BW), max 7.125GHz frequency, 28.21Gb/s transceiver, 67.8Mb memory
4-7GHz Patch Antennas	SMA	SMA	Resonant only on specific bandwidth

The RFSoc Kit includes a breakout board with ADC/DACs and SMA ports.

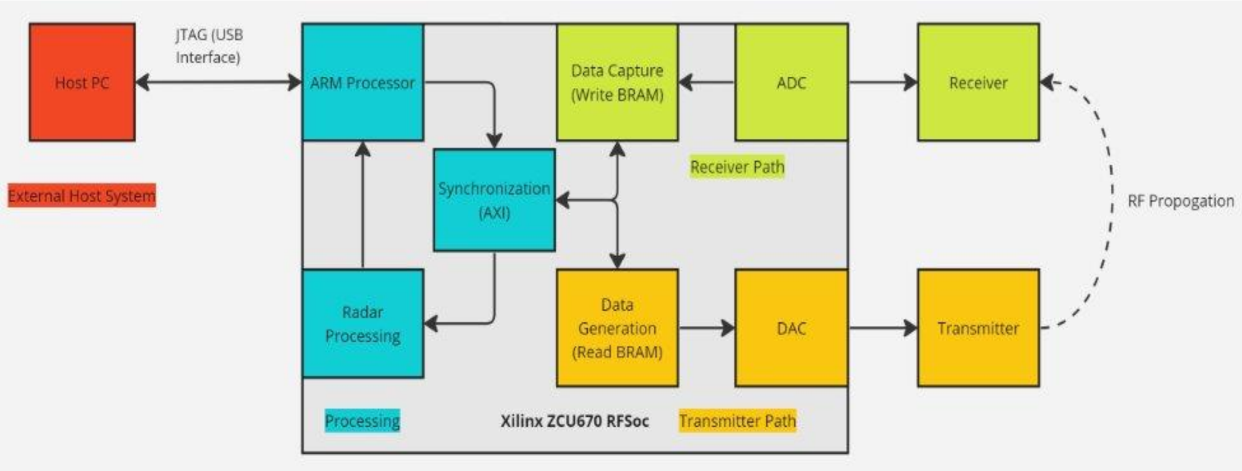


Figure 1 - Overall system design

Figure 1 outlines what parts of the RFSoC will be needed to accomplish this project. Each of the sections inside the Xilinx ZCU670 RFSoC rectangle represent different blocks in the Vivado design environment that directly interface with physical components on the RFSoC.

Software

The software required for this project is as follows in Table 2:

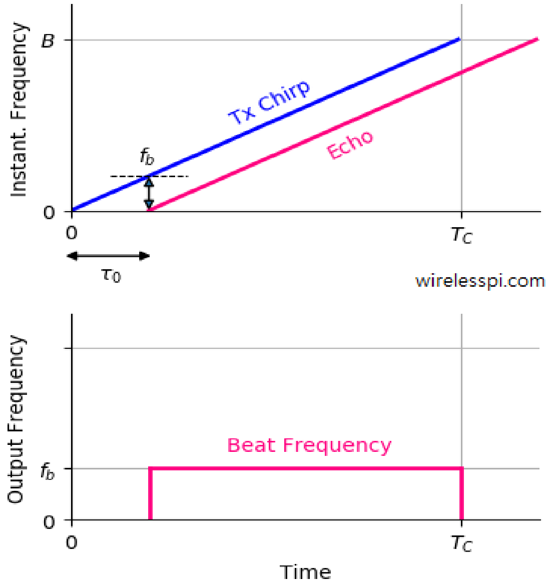
Table 2

	Inputs used	Outputs used	Constraints
MATLAB / Simulink	Parameters for chirp rate, BW, freq range	HDL code	This specific RFSoC is not supported in MATLAB libraries
Vivado	Board configuration (ZCU670), Vitis SDK (software configuration)	USB JTAG Serial (Heatmap/ control)	Vitis must use C++, there are official RfSoc Limitations in reference [1]

The role of MATLAB/Simulink in this project has been primarily for modeling DSP algorithms. There is no MATLAB developer support for the ZCU670 RFSoC so the only way to convert Simulink models to HDL (the hardware level language the RFSoC can be programmed with) is to use the HDL coder tool in Simulink without a target device. The HDL is used in Vivado as a custom IP.

Digital Signal Processing (DSP)

A frequency modulated continuous wave (FMCW) chirp signal is used as the transmit waveform for this project. To detect distance to an object, transmit a signal linearly increasing in frequency (figure 2). The signal will reflect off the object to be detected at some delay related to the speed of light. The frequency difference between TX and RX is directly related to the delay, which determines how far away the detected object is.



wirelesspi.com

Figure 2 - Chirp signal - linearly increases in frequency over time

A summary of the DSP for the RX waveform is as follows:

1. Sample output of the analog to digital converter (ADC)
   - This is the baseband frequency difference signal
2. Compute FFT to determine the beat frequency (difference in frequency)
   - Larger chirp bandwidths yield finer range resolution

$$\Delta d = \frac{c}{2B}$$

   - Higher sampling and longer chirps yield further max range

$$d_{max} \leq \frac{cT_c}{2B} F_s$$

3. Convert measured frequency to corresponding distance

$$d_{target} = \frac{cf_b T_c}{2B}$$

Specifications

The specifications for the product are as follows in Table 3, labeled by FUNCTION, PERFORMANCE, COMPLIANCE, ENVIROMENTAL, RELIABILITY, INPUT/OUTPUT, and CONFIGURATION:

Table 3	
FUN-1	System is able to determine distance of a target using radar and processing on the FPGA
FUN-2	Transmitter/Receivers/Antennas must be working in the 4-8 GHz range in the C band

PER-1	Digital signal processing subject to RFSoC specifications
PER-2	Should be able to have sub 3-meter detection
COM-1	Must operate in the ISM band
COM-2	Transmitter must keep power below 1W to prevent interference
ENV-1	Function within lab-based environment
REL-1	Must be able to be used repeatedly
I/O-1	Appropriate antennas for chosen frequency range and bandwidth.
I/O-2	Antennas will be directive
CON-1	Will be ground based, 1 antenna for transmit and receive, Using RFSoC for DSP and as a transceiver

Test Cases

Detailed reference to the test plans can be found in “Appendix A: Verification Test Plan”.

Test cases were identified for the purpose of verifying the capabilities and requirements of the design as pertaining to the specifications requested by the customer at the time of designing our product. Such test were to focus on the areas of resolution, power, object velocity identification, and antenna Standing Wave Ratio (SWR).

Build & Test

Project status/results

The status of the S24-41 project is still under development to reach full capabilities and requirements as outlined by the customer. As of this writing, the project is in a state of being able to transmit a signal that is increasing in frequency over time (a sawtooth signal) (figure 3) from the Xilinx RFSoC Evaluation Board. The Xilinx RFSoC Evaluation Board is also able to receive the same signal as the transmitted signal and display the result on a screen for visual interpretation (figure 4 and 5). Both transmitting and receiving can be done through RF propagation using the on-board ADCs (Analog to Digital Converter) and DACs (Digital to Analog Converter), along with a pair of resonate antennas. The current method to generate a transmitted signal and visualize a received signal is using the Xilinx RF Analyzer debugging tool, made specifically for Zynq UltraScale+ RFSoC products [2].

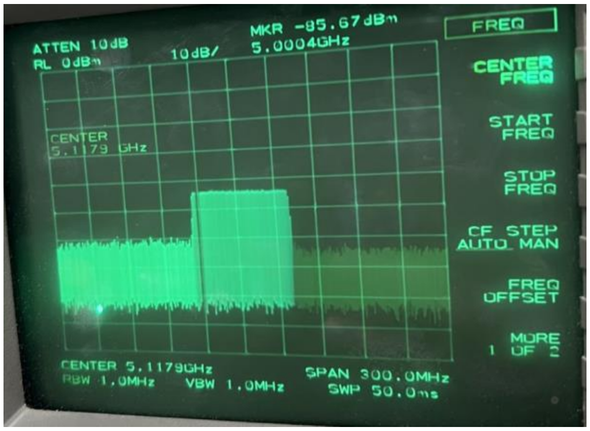


Figure 3 - Chirp signal as seen transmitted though the RFSoC DAC and shown on a spectrum analyzer

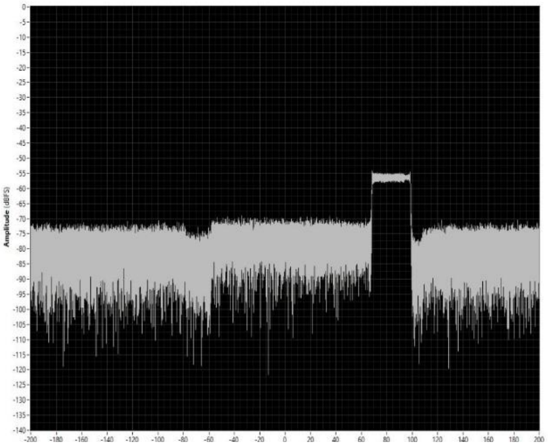


Figure 4 - Output of the RFSoC DAC

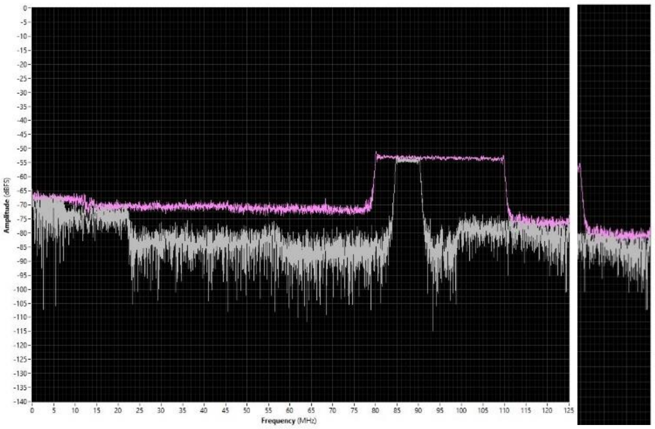


Figure 5 - Signal recieved by the RFSoC ADC with max hold

Figure 4 - Received signal with maxhold value (pink line)

Another key component of our project is creating a radar processing module in Simulink. This radar processing chain allows us to compute the distance from the board to an object in real time. In its current state, the Simulink model is not fully HDL-compatible, meaning that our FFT block is unable to be transformed into HDL and be uploaded to our board. This prevents us from deploying our custom signal processing chain onto the board to gather real world results.

Test results and analysis

In order to validate the Simulink system model, we created a MATLAB testbench program. This MATLAB program simulates the FPGA receiving our signal over a noisy channel and then uses FMCW signal processing to compute a distance estimate. This distance estimate calculated in MATLAB should be consistent with the Simulink model output to validate our model. It is important to emphasize that one of the major steps in the development of the Simulink model is converting stages to "'fixed point", which is the process of fixing how many bits are allocated to each signal. Therefore, the below test is important as it shows that despite using limited bit precision in our model, it is still capable of producing accurate outputs.

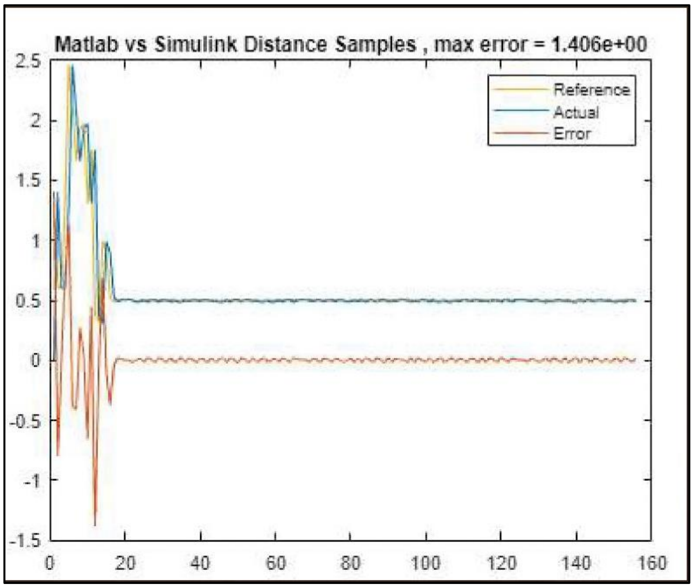


Figure 6 - Matlab Reference vs Simulink Actual

With this graph, for the majority of the test, our error is small demonstrating that our Simulink model is functioning consistent with the MATLAB reference. It is worth noting that the discrepancies in the beginning corresponded to noise when the object is not yet detected. This results in the output of the model outputting noise and thus we need not be concerned if there are differences between the model and our test bench as no object is detected at this phase.

Key accomplishments

- Design and simulate Simulink model for our custom signal processing chain, allowing us to receive distance to simulate objects with some real-world variables like random noise.
- Modified existing RF analyzer design to allocate additional BRAM resources to ADC and DAC signal blocks, allowing us to have a larger sample size for custom signal generation and higher fidelity received signal processing.
- Created board Linux image to allow the board to be self-deployed and not connected to host computer.
- Created additional documentation to assist future students with setup and configuration of the board

Delivery

Project Performance

We made significant progress toward our goals but were unable to deliver a fully working project within the course's timeline. This was largely due to our newly released FPGA being incompatible with the Simulink HDL code generation toolkit. Despite this roadblock, we were still able to develop proof of concept models that simulate our system virtually and demonstrate that it is within the capabilities of our hardware once more support becomes available for the board. Additionally, we were able to transmit and receive our desired signals by using tools native to the board further proving the feasibility of this project once we are capable of uploading the signal processing portion of the system to the board.

Customer Satisfaction

Our customer, Dr Sarah Jenson and the team have had numerous meetings throughout the design process. The purpose of these meetings was to update her on our progress. We have had to update our customer with our refined set of deliverables as we gained a better understanding of project scope and feasibility.

Deliverable Status

The deliverables for our project are internally managed via Microsoft Teams and are currently publicly available on Github.com at the following url: https://github.com/xXallisonXx/MDE-S24-41. Our documentation includes the MATLAB testbench programs, our Simulink models of the system, and the HDL code we generated from our Simulink model. Additionally, we have uploaded some miscellaneous MATLAB files we used to generate chirps using the RF Analyzer tool provided with our development board. Lastly, we have uploaded our poster and PowerPoint presentation materials we plan to deliver during our engineering expo covering our project.

Challenges / Issues

Most challenges encountered during the process of this project were centered around the availability of 3rd party support and documentation. During the process of this project the Zynq UltraScale+ RFSoC DFE ZCU670 Evaluation Kit was still considered to be in an “early-access” status by the manufacturer Xilinx / AMD. The result of this status means that there was limited publicly available documentation available and that in order to obtain some specific documentation as well as software licenses, our team was required to submit a request for access to the “RFSoC DFE Document Lounge” and the ”RFSoC Tools Lounge.” The most important parts of these lunges are the documents about the board and Xilinx software like the RF Analyzer (located in the documents lounge) and the license provided in the tools lounge, to allow a user to generate bitstreams for the ZCU670 board.

Subsequent challenges of a lesser degree were encountered when our team was ready to convert the signal processing code that was implemented in MATLAB and Simulink into a Hardware Description Language (HDL) that could be implemented by the Field Programmable Gate Array (FPGA). The issue arising was partially due to the aforementioned “early access” status of the board but was also in part due to the requirements of the “HDL Coder” application in MATLAB. The HDL Coder generally has support for various FPGA products, however due to the newness of our FPGA there was no specific board support, to overcome this the team made a generalized version of the code that was not dependent on-board specific Inputs and Outputs (I/O). By using a more generalized model in Simulink, we could progress further into HDL development.

The next issue is currently unresolved but is believed to be relatively trivial to fix given a bit more time. Is the implementation of a custom Fast Fourier Transform in Simulink compatible with the HDL Coder application? The one originally used was not compliant with the current use of HDL Coder, but there are support forums that detail a readily available solution to this issue. However, these possible solutions have not been thoroughly tested in our current implementation due to time constraints.

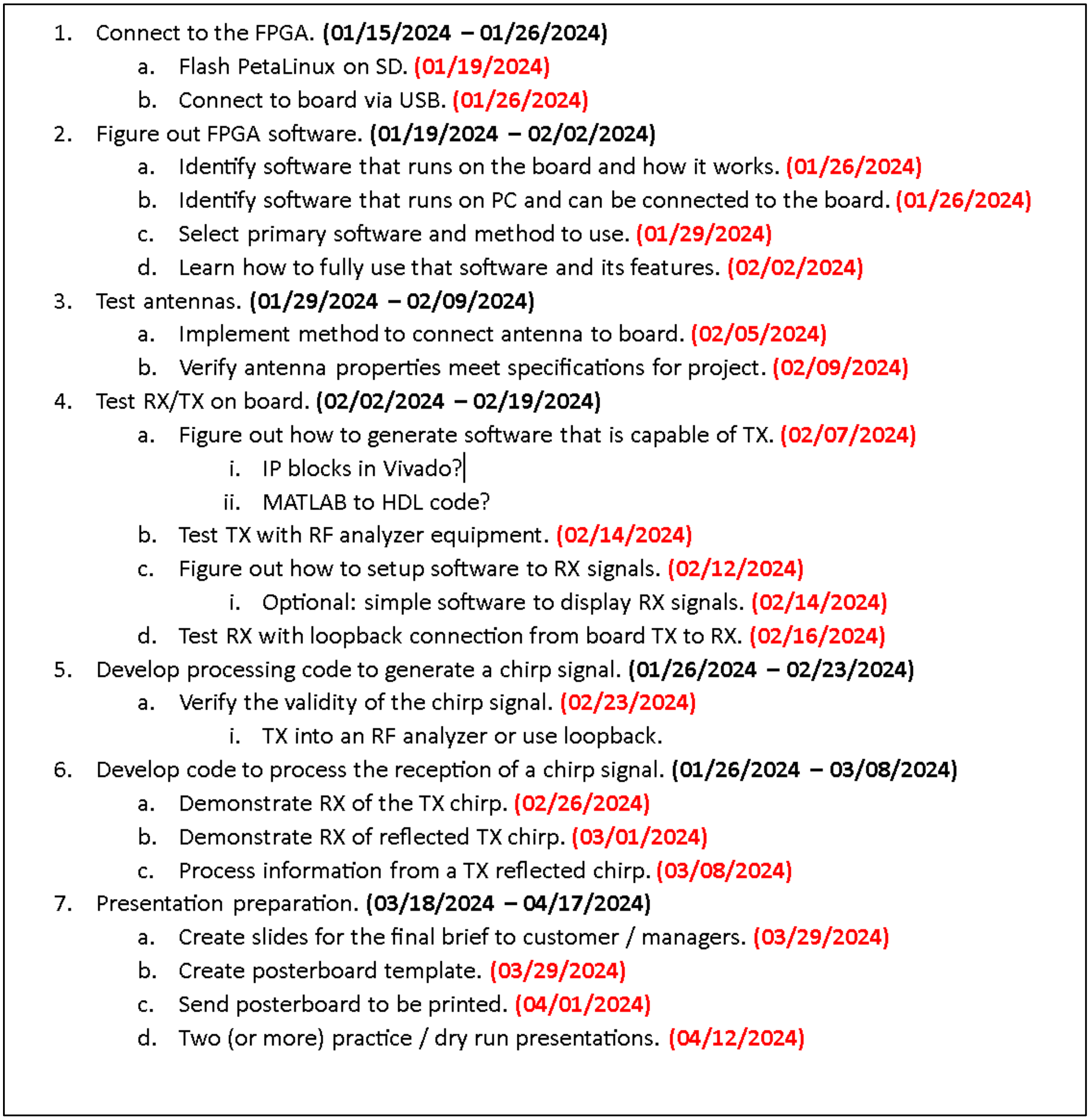
Note that Xilinx Vivado is the only hardware design tool currently supported on the ZCU670 platform. Several other Xilinx boards, including RFSocs like the ZCU111 have publicly available MATLAB support. This MATLAB support provides an abstraction layer away from the hardware specifics of Xilinx Vivado, and has several publicly available tutorials to assist a team with developing a radar system. We tried defining a custom board plugin in the SoC Blockset extension for MATLAB/Simulink, but this proved to be difficult since our board has several proprietary parts that were not natively supported in the board definition. Thus, we would have had to define these new parts, which proved to be inconsistent and expensive, to the point where that idea was scraped. Instead, we used the aforementioned Vivado Design Suite to create our modified bitstreams for use with the RF Analyzer tool.

Schedule and Cost

The schedule followed for this project was based on the time availability of the team with considerations given heavily on the requirements of each member to other required obligations such as class, assignments, exams, and other duties that would normally not be an issue for a team of full-time employees. Figure 5 demonstrates the scheduled used during the 2nd part of the 2-semester course used to develop this project. Some tasks have similar deadlines (indicated in red), these tasks were capable of being completed in parallel and were not dependent on prior tasks.

Costs for our project are based mainly on necessary equipment due to no accurate way to confidently calculate total worked hours that separate time spent on course materials and time spent on direct project related task. The project had 2 main sources of costs:

1. Zynq UltraScale+ RFSoC DFE ZCU670 Evaluation Kit ($12,954.00)
2. 8x 4.9GHz ~ 5.925GHz antennas (~$16.00)



1. Connect to the FPGA. **(01/15/2024 – 01/26/2024)**
    a. Flash PetaLinux on SD. **(01/19/2024)**
    b. Connect to board via USB. **(01/26/2024)**
2. Figure out FPGA software. **(01/19/2024 – 02/02/2024)**
    a. Identify software that runs on the board and how it works. **(01/26/2024)**
    b. Identify software that runs on PC and can be connected to the board. **(01/26/2024)**
    c. Select primary software and method to use. **(01/29/2024)**
    d. Learn how to fully use that software and its features. **(02/02/2024)**
3. Test antennas. **(01/29/2024 – 02/09/2024)**
    a. Implement method to connect antenna to board. **(02/05/2024)**
    b. Verify antenna properties meet specifications for project. **(02/09/2024)**
4. Test RX/TX on board. **(02/02/2024 – 02/19/2024)**
    a. Figure out how to generate software that is capable of TX. **(02/07/2024)**
        i. IP blocks in Vivado?
        ii. MATLAB to HDL code?
    b. Test TX with RF analyzer equipment. **(02/14/2024)**
    c. Figure out how to setup software to RX signals. **(02/12/2024)**
        i. Optional: simple software to display RX signals. **(02/14/2024)**
    d. Test RX with loopback connection from board TX to RX. **(02/16/2024)**
5. Develop processing code to generate a chirp signal. **(01/26/2024 – 02/23/2024)**
    a. Verify the validity of the chirp signal. **(02/23/2024)**
        i. TX into an RF analyzer or use loopback.
6. Develop code to process the reception of a chirp signal. **(01/26/2024 – 03/08/2024)**
    a. Demonstrate RX of the TX chirp. **(02/26/2024)**
    b. Demonstrate RX of reflected TX chirp. **(03/01/2024)**
    c. Process information from a TX reflected chirp. **(03/08/2024)**
7. Presentation preparation. **(03/18/2024 – 04/17/2024)**
    a. Create slides for the final brief to customer / managers. **(03/29/2024)**
    b. Create posterboard template. **(03/29/2024)**
    c. Send posterboard to be printed. **(04/01/2024)**
    d. Two (or more) practice / dry run presentations. **(04/12/2024)**

Figure 5: Schedule for second half of 2-semester MDE course.

Lessons Learned

Because each team member can have wildly varying lessons learned from their unique project perspective, the bullet points below list the lessons learned in no particular order.

- Time management can be challenging when each member has their own schedule.
- Having a dedicated and consistent meeting time every week can help provide team stability and accountability.
- Don’t be afraid to ask for help/advice. Your teammates, mentor, SME, and customer all want to see you succeed and are great resources for help.
- Starting early is always a good idea if you can. We took a while getting the board setup and deciding how to approach programming the board. This delay pushed our progress back, leading us to have two separate parts for our deliverables rather than a unified deliverable.

References

[1] Xilinx, "71746 - Zynq UltraScale+ RFSoC: RF Analyzer Known Issues and Limitations," September 2021. [Online]. Available: https://support.xilinx.com/s/article/71746?language=en_US. [Accessed 2024].

[2] Xilinx, "RF Analyzer," 11 June 2021. [Online]. Available: https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/808419337/RF+Analyzer.

Appendix A: Verification Test Plan

Team #:	41	**Team Name:**	SARDINE
Date:	4/15/2024 16:29	**Document Owner:**	Thomas Skidmore
Revision #:	1		

Subsystem/ Function/ Feature Name:	Radar Resolution	**S1 Test Plan**
Date of Test:	10/18/2023	
Performed By:		
Tested By:		
Concluded Condition of meeting Engineering Specification:	**Compare resolution at different distances.**	

Description: Radar resolution is an overall performance metric of our system. The resolution is a measurement of how sensitive the radar system is to change in target distance. For example, a 3 meter resolution system can be assumed to reliably create an accurate new distance to target (± 10% of total distance).
I/O: Inputs include power cable, radio receivers. Outputs include calculated distance to target, and transceivers.
Safety: Avoid direct contact with radio antenna during transmission.

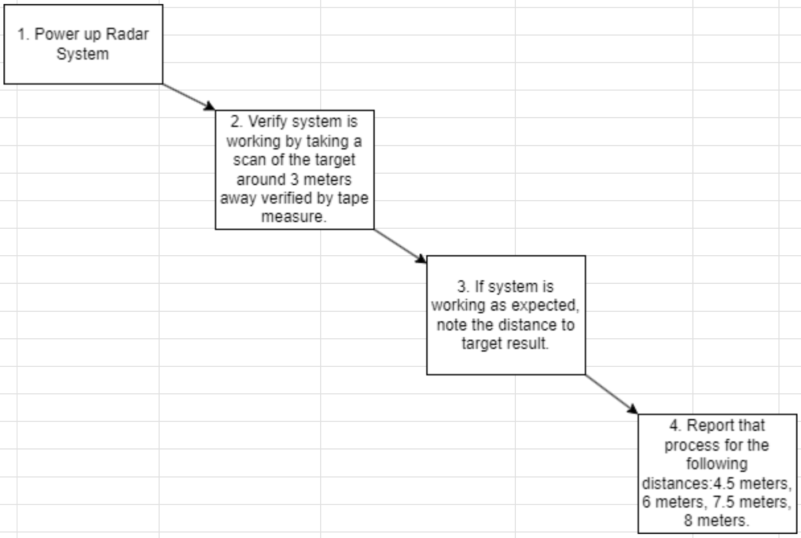
Specification Number	Importance	Source	Function	Specification (Metric)	Unit of Measure	Marginal Value	Ideal Value	Comments/Status
PER-2	1	Requirments	System Performance	Resolution of System	Meters	0-3 m	1.5 m	

II. EQUIPMENT REQURIED

Specification Number	Equipment or Instrumentation required			
PER-2	Tape Measure	Target	Radar System	120V AC Outlet

III. DATA COLLECTION STRATEGY

Specification Number	Data acquisition strategy
PER-2	1. Power up Radar System 2. Verify system is working by taking a scan of the target around 3 meters away verified by tape measure. 3. If system is working as expected, note the distance to target result. Take three more snapshots (4 total). 4. Report that process for the following distances: 6 meters, 9 meters, and 12 meters.



1. Power up Radar System

2. Verify system is working by taking a scan of the target around 3 meters away verified by tape measure.

3. If system is working as expected, note the distance to target result.

4. Report that process for the following distances:4.5 meters, 6 meters, 7.5 meters, 8 meters.

Team #:	P17431	**Team Name:**	SARDINE POWER LIMITS TESTING
Date:	10.19.23	**Document Owner:**	Elliott Kroll
Revision #:	1		

S1 Test Plan

Subsystem/ Function/ Feature Name:	
Date of Test:	Antenna/ Power Limit: SARDINE Power Limit verifification
Performed By:	
Tested By:	

Concluded Condition of meeting Engineering Specification:

I. TESTING SPECIFICATION

Specification	Importance	Source	Function	Specification (Metric)	Unit of	Marginal	Ideal	Comments/Status
S1	1	FCC Requirements		Power	Watts	.+/-0.01 W	1>	

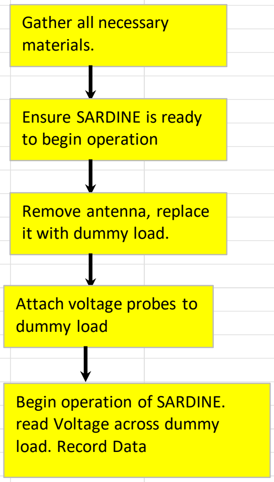
II. EQUIPMENT REQURIED

Specification	Equipment or Instrumentation required
S1	Dummy Resistor (500K ohm), Oscope

III. DATA COLLECTION STRATEGY

Specification	Data acquisition strategy
S1	When SARDINE is in operation, a dummy load will be attached in place of the antenna. From there, the power of device can be measured by mesauring the voltage across the dummy load. The power equals (V^2)/R where R is the resistance of the dummy load.

III. TESTING FLOWCHART



Gather all necessary materials.

Ensure SARDINE is ready to begin operation

Remove antenna, replace it with dummy load.

Attach voltage probes to dummy load

Begin operation of SARDINE. read Voltage across dummy load. Record Data

Team #:		**Team Name:**	SARDINE
Date:	10/18/2023 0:00	**Document Owner:**	Paul Broome
Revision #:	1		

PER-2 Test Plan

Subsystem/ Function/ Feature Name:	Velocity detection of slow (< 2m/s) moving objects
Date of Test:	
Performed By:	
Tested By:	

Concluded Condition of meeting Engineering Specification:	

I. TESTING SPECIFICATION

Specification Number	Importance	Source	Function	Specification (Metric)	Unit of Measure	Marginal Value	Ideal Value	Comments/Status
PER-2	1		System	Vmeasured-Vactual	m/s	50%	0	

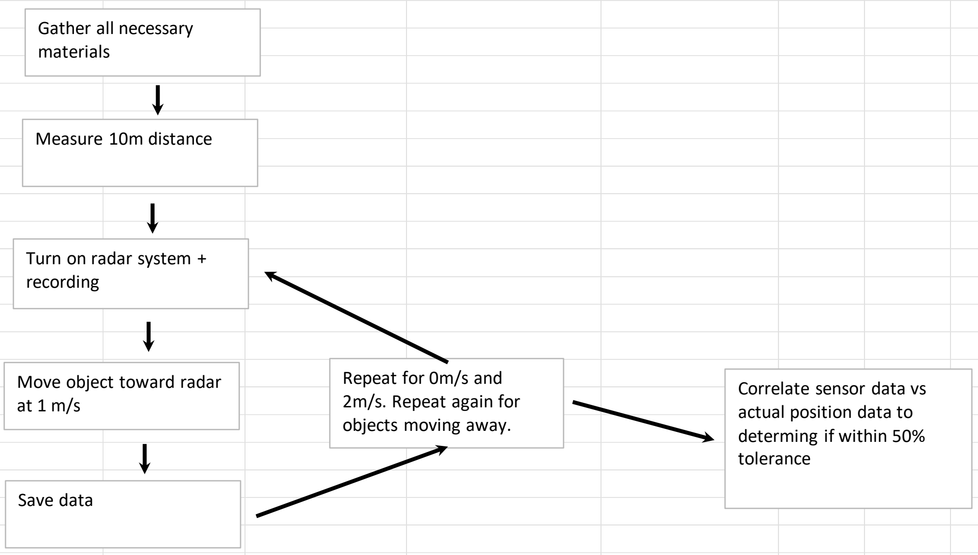
II. EQUIPMENT REQURIED

Specification Number	Equipment or Instrumentation required
PER-2	stopwatch, tape measure.

III. DATA COLLECTION STRATEGY

Specification Number	Data acquisition strategy
PER-2	The goal of this test is to determine how accurately the radar system can measure the velocity of an object travelling directly toward the radar system. The system should be capable of accuratley measuring the velocity of objects towards or from the system at speeds less than 2m/s. There are no major safety concerns with this test.

III. TESTING FLOWCHART


Gather all necessary materials
Measure 10m distance
Turn on radar system + recording
Move object toward radar at 1 m/s
Save data
Repeat for 0m/s and 2m/s. Repeat again for objects moving away.
Correlate sensor data vs actual position data to determing if within 50% tolerance

Team #:	41	**Team Name:**	SARDINE
Date:	4/15/2024 16:31	**Document Owner:**	Vincent Snell
Revision #:	0		

Test Plan

Subsystem/ Function/ Feature Name:	Standing Wave Ratio of Antennas
Date of Test:	TBD
Performed By:	Vincent Snell and Team
Tested By:	Vincent Snell and Team

Concluded Condition of meeting Engineering Specification:

I. TESTING SPECIFICATION

Specification	Importance	Source	Function	Specification (Metric)	Unit of	Marginal	Ideal	Comments/Status
S1	3	Antenna	Transmitter	Standing Wave Ratio	SWR	1 + 0.7	1	

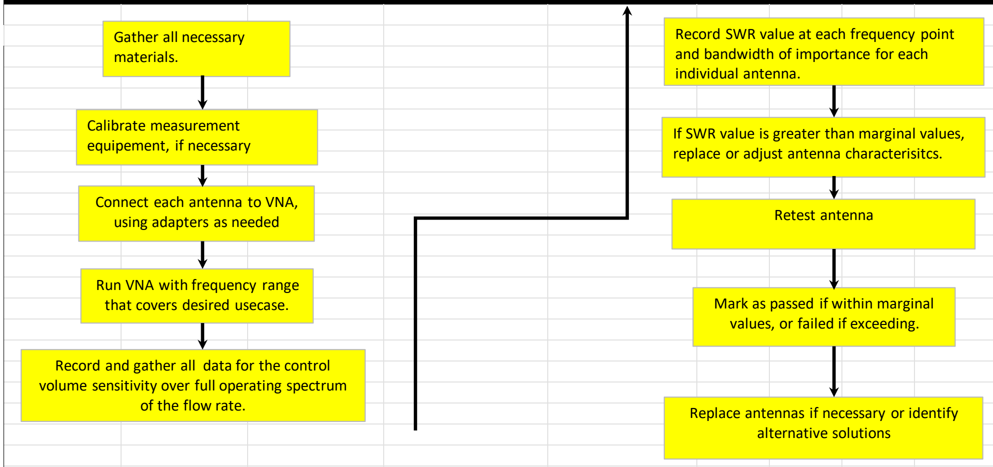
II. EQUIPMENT REQURIED

Specification	Equipment or Instrumentation required
S1	Antenna Analyzer or Vector Network Analyzer, Antenna, Coaxial Cables, Attenuators, Connector Adapters

III. DATA COLLECTION STRATEGY

Specification	Data acquisition strategy
S1	Purpose of this test is to properly measure the efficency of the antennas used for transmission and reception of radio frequencies by obtaining the resonant values of the antennas. The more resonant an antenna is for our use case the higher the potential for more accurate and repeatable results.

III. TESTING FLOWCHART



Gather all necessary materials.

Calibrate measurement equipement, if necessary

Connect each antenna to VNA, using adapters as needed

Run VNA with frequency range that covers desired usecase.

Record and gather all data for the control volume sensitivity over full operating spectrum of the flow rate.

Record SWR value at each frequency point and bandwidth of importance for each individual antenna.

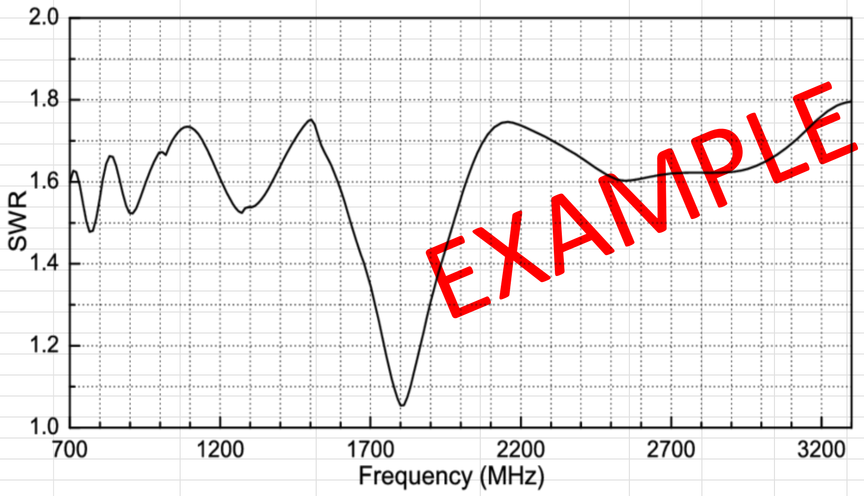
If SWR value is greater than marginal values, replace or adjust antenna characterisitcs.

Retest antenna

Mark as passed if within marginal values, or failed if exceeding.

Replace antennas if necessary or identify alternative solutions

IV. RAW DATA ACQUISITION


SWR
Frequency (MHz)
EXAMPLE