Preliminary Detailed Design Report (Version 1)

FPGA Enhanced Digital Beamsteering Phased Array

Team 311: Katheryn Potemken, Tiernen Pan, Christian Balos, William Snyder, and Andrew Cayson

I. Introduction

Problem Statement: The team needs to develop a transmitting phased antenna array that can control the main lobe of radiation by controlling the digital baseband signal.

Motivation: The motivation for beamsteering is the need for higher data transmission rates. It allows for transmission of higher quality signals to receivers. This means that the Signal to Noise Ratio is significantly higher leading to fewer errors in the transmission of data. Another upside to using beamsteering is that there is no need to increase the transmitting power in order to achieve the higher quality signal. By focusing the main lobe of the transmission radiation, we are also able to decrease the amount of interference that we inflict on other receivers because of the large attenuation in the sidelobes that comes as a by-product of beamsteering.

Requirements: The system must include an antenna array, RF up-conversion channels, digital to analog converters, and an FPGA with the optimum radiation beam control algorithm. The operating frequency needs to be within the ISM band and the output power that leads into the antenna must be less than 30 dBm. The steering angle of the beam needs to be able to have a 180 degree range.

II. Selected Concept

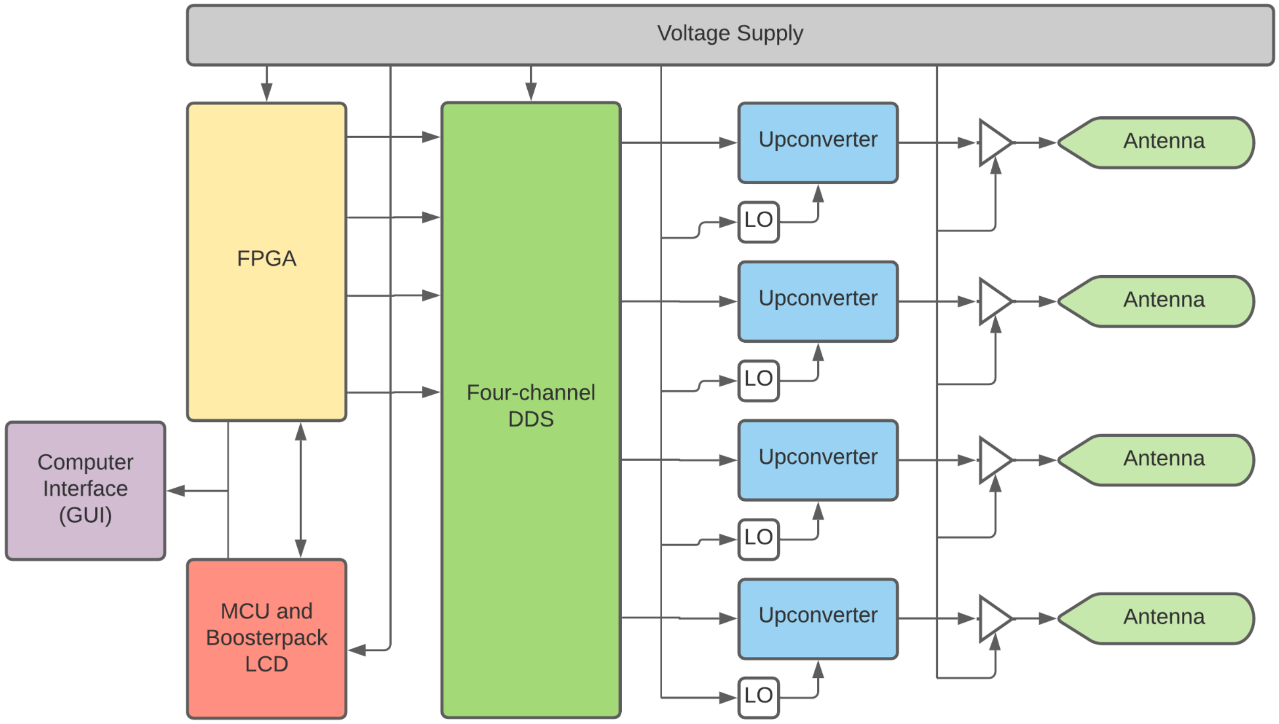
When selecting final concepts, the first main decision was to select the type of system. Various types of systems are digitized, digital, and analog. The team has selected the digital route to implement the beam steering system. This will allow us to initially create and control the signal digitally, which later be converted to an analog signal.

We will implement both the FPGA and MCU within our design. The FPGA allows us to program and control the functionality of our system. The MCU allows us to take advantage of the booster pack which offers us many extensive capabilities which will come in handy when considering the computer interface/GUI.

The size of the system will be handheld size. That way it can be easily transported to areas of high signal density, which may also help us when testing our final design.

The system will initially be powered through USB port connections from our laptops. The USB connection will also allow us to upload our programs to the FPGA and MCU. Once the functionality of our system works properly, we will then implement a power source. An external power DC power supply with USB port connection will be a reliable power source to ensure power will be constantly supplied through our system.

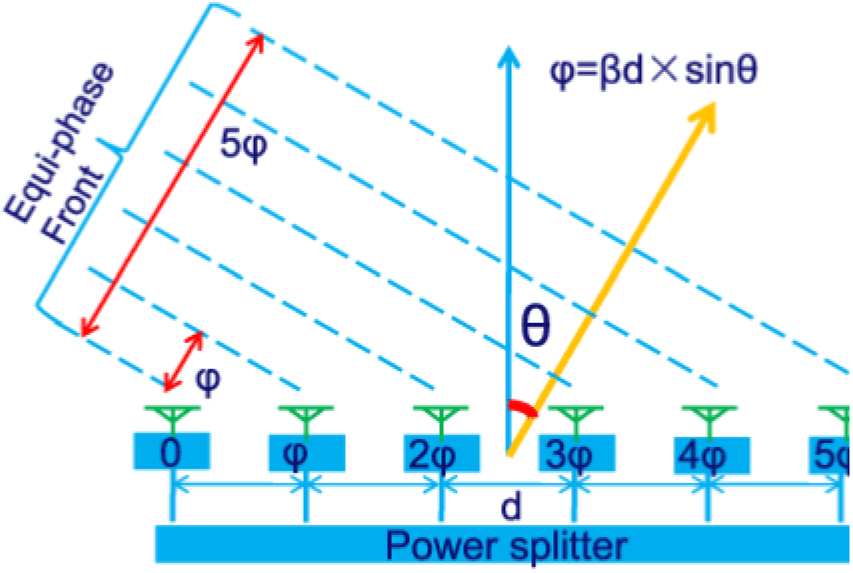
III. Preliminary Design



Voltage Supply: As previously stated, the system will initially be powered through USB port connections from our laptops. This allows us to properly upload and load our programs to the FPGA and MCU. Once we can ensure our system works as desired and the functionality of the system is perfected, we will then implement an external power source. An external power DC power supply with USB port connection will be a reliable power source to ensure power will be constantly supplied through our system. We want a reliable power source because if the power supplied to the system is interrupted at any point and time, it will affect the functionality of the system. Thus, a reliable constant power source will lead to a reliable design.

Computer Interface (GUI): The system GUI will be controlled through various software on laptops. For programming the FPGA, we will use the software Quartus. Quartus is a programmable logic device design software which enables analysis and synthesis of HDL designs. Here we will write a series of VHDL programs to run the system. For programming the MCU, we will use the software code composer studio (CCS). CCS is an integrated development software composed of tools used to develop and debug embedded applications. Here we will use C to write a series of programs for the user interface. The booster pack on the MSP430 will allow the user to interact and control the system independently from the laptop.

FPGA/MCU: For the project, the team is intending to use the Xilinx Spartan 6 XC6LX16-CS324 FPGA. The microcontroller is going to be the Graphical User Interface (GUI) that the user is able to interact with. The FPGA will take in the angle that the user wants the beam to point. The FPGA will generate a clock cycle and this output feeds into the Direct Digital Synthesizer (DDS), which will generate a sinusoidal signal for each of the channels. The FPGA will also compute the phase shift angle that the DDS needs to shift based on the input angle that the user wants the beam to point. A basic diagram relating the equation and the direction of the beam is shown below:



Where beta is 2 times pi divided by the wavelength of our frequency, which is 2.4 GHz.

Four-Channel DDS: The four channel DDS will be an AD9959 development board that consists of four direct digital synthesizers. The board will receive a reference clock signal from the FPGA that will dictate the amplitude of the base signal the four DDSs will use. Each DDS will have a separate input signal that will correspond with the required phase delay needed for each signal. The module will output four signals that will need to be amplified before being input into the antennas.

AD9959/PCBZ

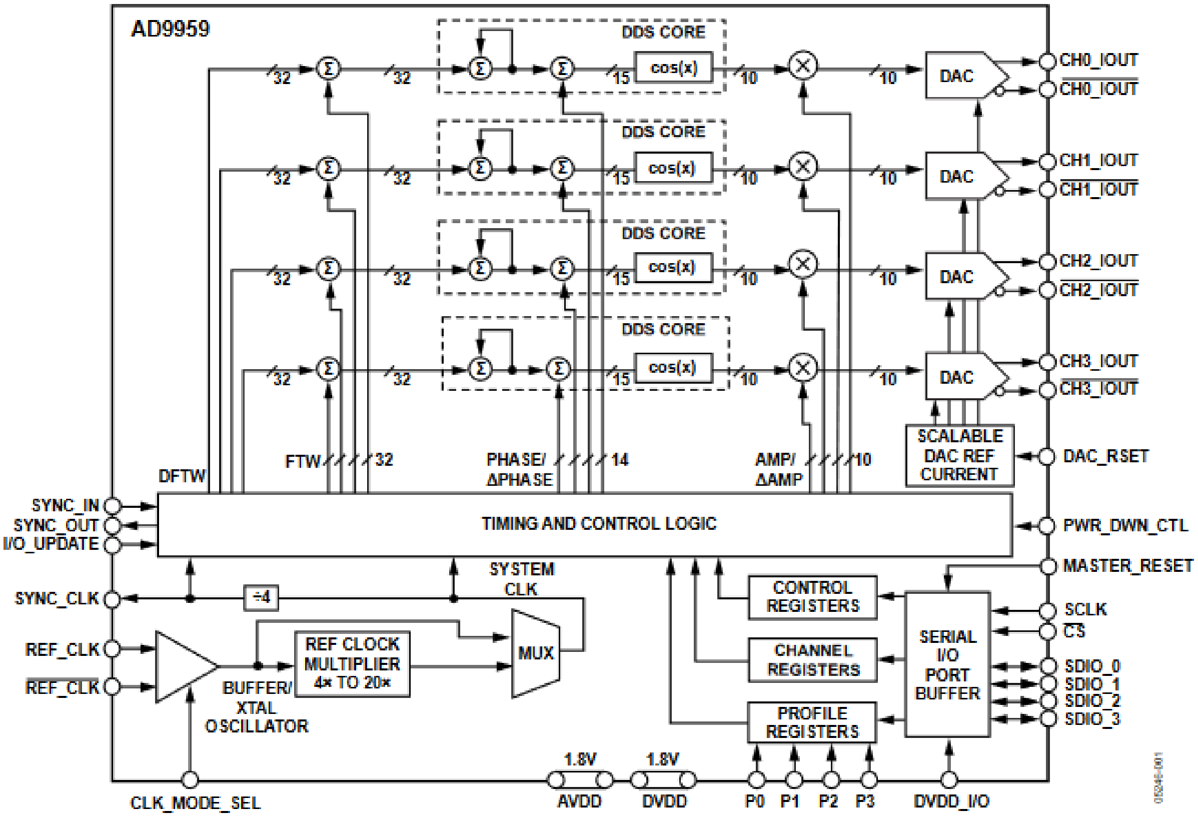
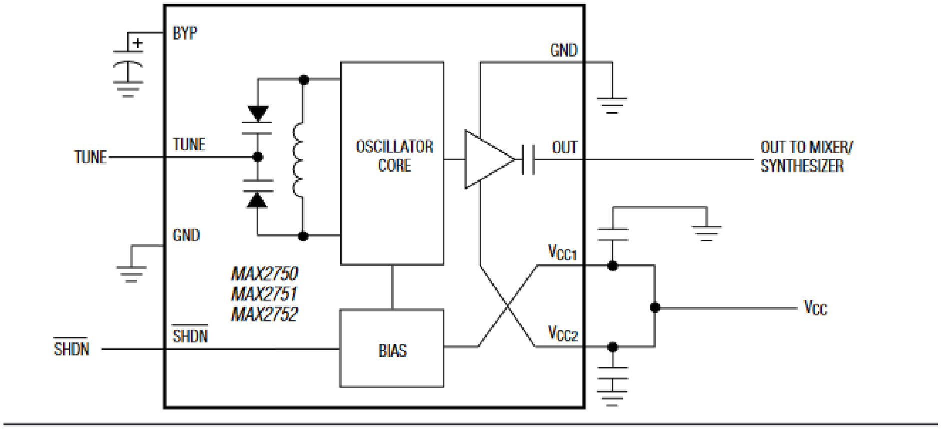


Figure 2. Detailed Block Diagram

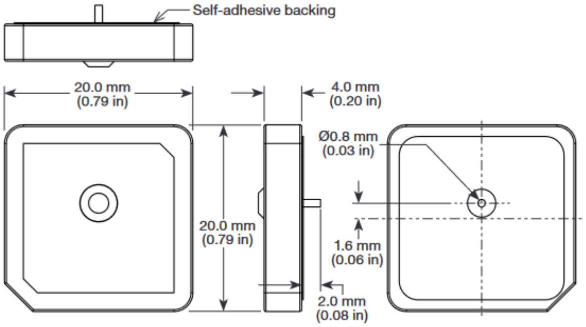
Up Converter: The upconverter will be a mixer that will combine the frequency of the signals from the DDSs with the frequency of a signal supplied by a local oscillator to reach the desired frequency.

Local Oscillator (LO): The local oscillator will be used to add the signal with the required frequency to the mixer/ upconverter. A voltage supply will supply the required input power



Amplifier Gain: A gain amplifier will be used to add power to the signal before inputting into the antenna. Power should be <30dBm.

Antenna: Four 2.4GHz patch antennas will be used. The antennas will be spaced 0.5 wavelength apart, which is about 2.46 inches from center at 2.4GHz. Since our functional decomposition we have decided to move away from using omnidirectional antennas to using directional patch antennas. Omnidirectional antennas would provide a second beam which is not needed four the purposes of this project. Directional patch antennas will be simpler to implement.



IV. Summary

The team has selected a digital system for the project. A brief breakdown of the components for the system: The GUI is the user interface, where the user can select/control the angle that the beam is pointing. The team selected a microcontroller with a boosterpack to be the GUI. The FPGA generates the clock cycles and the phase delays required for the DDS to correctly generate four signals that are offset in a way that the beam points in the direction the user intended. The upconverter takes the input from the FPGA, generates the offset signals, and converts them from digital signals to analog ones. The mixer takes the output from the DDS and the output of a local oscillator and combines them to create the designated frequency (around 2.4 GHz). The signal out of the DDS is then fed into an amplifier that adjusts the output power so that it is acceptable for the antenna.The antenna outputs the voltage based signal into a radio signal.