Team 303: Software Radio Transmitter

Team Members: Kira Bronstein, Simon Charry, Christian Pollock, Jaryd Walton, Evan Woodard

FAMU-FSU Engineering

NORTHROP GRUMMAN

Abstract

Radios have been used for many generations to send and receive data over large distances, wirelessly. From the military and large corporations to private owned vehicles, radios are used in a variety of different ways. Oftentimes, companies or the military use many different radios for different tasks. All these different radio transmitters can add up to a larger required investment. However, a new radio on the cutting-edge of technology can be used to decrease these costs - the Software Defined Radio (SDR). An SDR is a radio that uses one single device while allowing software to control the radio. Instead of using different radios, different programs can be written to the same device. This project focuses on developing an SDR with three objectives. The radio is built from commercial parts, the software is reprogrammable, and the SDR must have high fidelity. This means that the signal reproduced will be of high quality. By creating an SDR, it allows businesses to be more cost-efficient and will allow anyone to create a design. The SDR will have only one radio, which will prevent any confusion that multiple radios can lead to. The SDR is useful in many areas because of its flexibility and easily maintained because of its use of commercial parts.

Background

- Radios provide easy and effective communication between users for many different applications.
- The introduction of software to deal with the filtering, amplifying, etc. normally taken care of by hardware components through the use of FPGAs or other computers saves time and money.
- The purpose of the project is to create a high-fidelity software-defined radio (SDR) with low-cost parts from different manufacturers.

Design Implementation

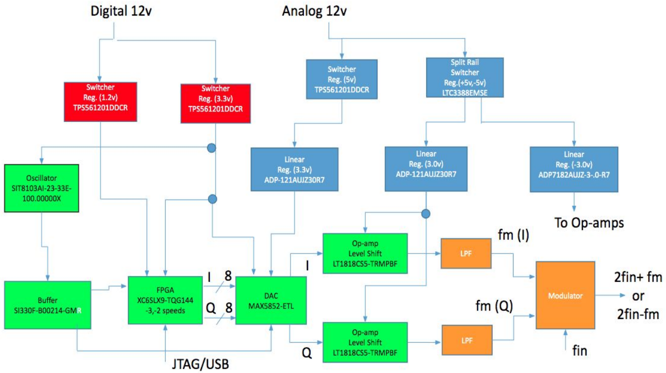

Figure 1: Block Diagram

Testing Procedure

FPGA/DAC:

1. Send data bit file which contains information of current level to be produced through the JTAG.
2. Determine the expected current produced based on the data bit file from the DAC.
3. Measure the voltage and calculate the current being produced and compare the expected with the measured.

Op-Amp:

1. Test the DC-offset of the Op-Amp by looking at the signal of the input and output and seeing if they change based on the value given at the DC-offset port.
2. Test the gain of the Op-Amp versus different frequency values based on the datasheet specifications (Figure 2).

Modulator:

1. Test the modulator circuit by using a frequency spectrum analyzer and compare to the ADS simulation

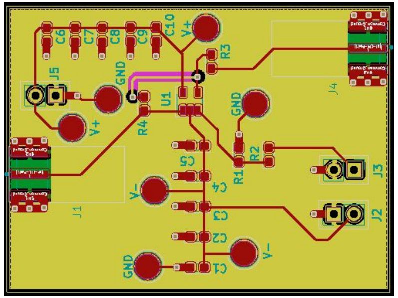

Figure 2: PCB Layout of Op-Amp from KiCad

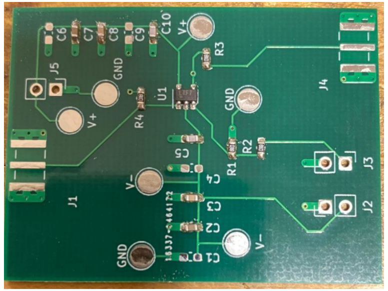

Figure 3: PCB of Op-Amp