FAMU/FSU College of Engineering

Department of Electrical and Computer Engineering

Preliminary Detailed Design

Team 303 Software Defined Radio

Names: Kira Bronstein

Simon Charry

Christian Pollock

Jaryd Walton

Evan Woodard

Date: December 2, 2020

I. Introduction

A. Problem Statement

Radios allow for the transmission of data and messages to a recipient at unprecedented speeds and distances. The increasing demand for cheap and effective electronic communications systems for various applications creates an interesting design problem of how to create a radio without so many hardware components; an SDR is a solution to just that. Due to their importance, many companies rely on radios for many different tasks. This, of course, would require multiple different radios to do said tasks. This is a problem for companies as radios are not cheap.

B. Motivation

This project takes on the task of developing an SDR through the use of readily available and cheap components to create an effective, versatile, and completely reprogrammable electronic communication system. The system must be easily reprogrammable as this is the main aspect companies are looking for, as it saves them tons of money in the long run by using and reusing one radio rather than buying multiple.

C. Requirements

- Design software (KiCAD, LTSpice, ADS, MATLAB)
- Digikey components (resistors, inductors, capacitors)
- Soldering Equipment
- PCBs
- Xilinx Spartan 6 FPGA
- ICs:
  - TPS561201DDCR (Switcher Regulator) x3
  - SIT8103AI-23-33E-100.00000X (Oscillator) x1
  - SI330F-B00214-GM (Buffer) x1
  - MAX5852-ETL (DAC) x1
  - LTC3388EMSE (Split Rail Switcher) x1
  - ADP-121AUJZ30R7 (Linear Regulator) x2
  - ADP7182AUJZ-3-.0-R7 (Linear Regulator) x1
  - LTC1818-CSS-TRMPBF (Op-Amp) x2
- In-house component assembly

II. Selected Concept

The final selection for the FPGA board of the design is Xilinx Spartan 6, XC6SLX9-2TQG144C. The Spartan 6 has a low cost, low power, and high I/O performance with small PCB footprint compared to other FPGA families. The final selection will also involve TPS561201DDCR for as Voltage Regulators, SI5330F-B00214-GMR for as buffer, MAX5852ETL+T for as DAC, and SIT8103AI-23-33E-100.00000X for as Oscillator. The software to design a PCB board and simulate the final schematic for final selection design are KiCAD and LTSpice. The component of voltage regulators, buffer, and oscillator provide the right specification of I/O for each module. These components also provided elements of familiarity, simplicity, and reliability.

III. Preliminary Design

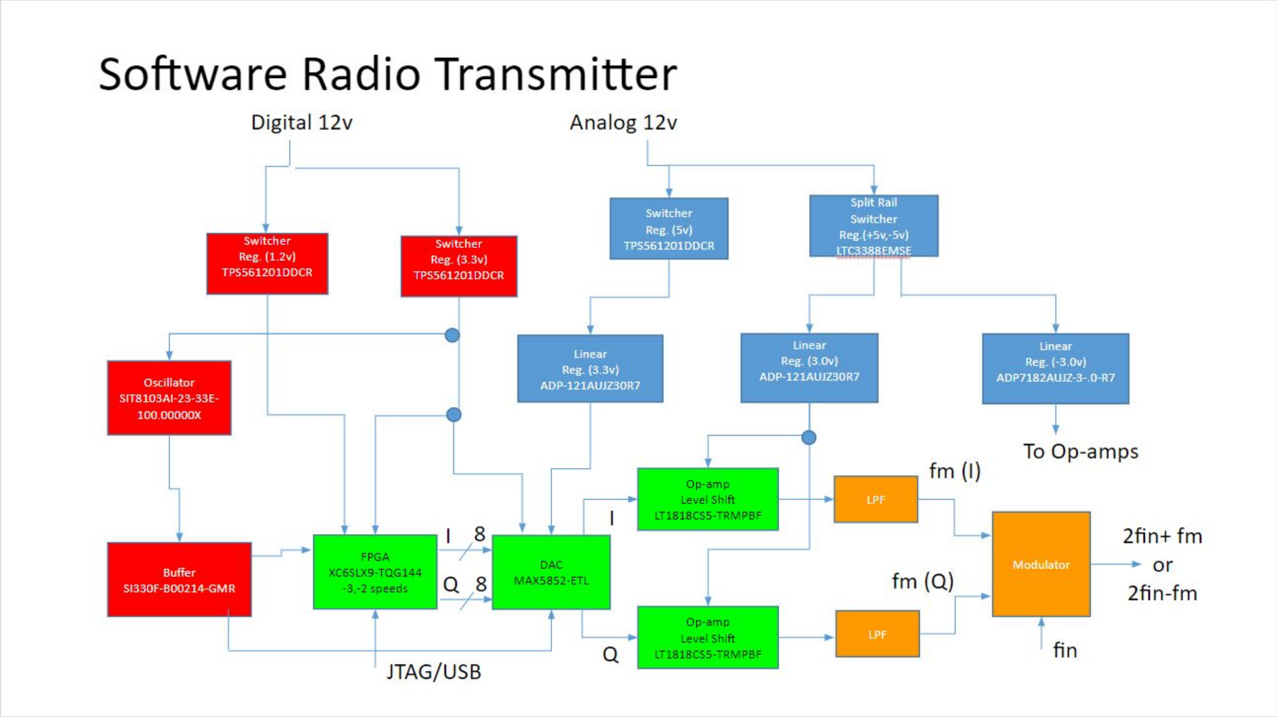
Software Radio Transmitter
Digital 12v
Analog 12v
Switcher Reg. (1.2v) TPS561201DDCR
Switcher Reg. (3.3v) TPS561201DDCR
Switcher Reg. (5v) TPS561201DDCR
Split Rail Switcher Reg.(+5v,-5v) LTC3388EMSE
Oscillator SIT8103AI-23-33E-100.00000X
Linear Reg. (3.3v) ADP-121AUJZ30R7
Linear Reg. (3.0v) ADP-121AUJZ30R7
Linear Reg. (-3.0v) ADP7182AUJZ-3-.0-R7
To Op-amps
Op-amp Level Shift LT1818CS5-TRMPBF
LPF
fm (I)
Buffer SI330F-B00214-GMR
FPGA XC6SLX9-TQG144 -3,-2 speeds
I
8
Q
8
DAC MAX5852-ETL
I
Modulator
2fin+ fm
or
2fin-fm
fm (Q)
Op-amp Level Shift LT1818CS5-TRMPBF
LPF
fin
Q
JTAG/USB

Figure 1: Detailed Block Diagram

I/O of each module (in tabular form)

	Sources (Digital and Analog)	Switcher Regulators	Linear Regulators
Input	N/A	12 V (Digital and Analog)	12 V (Digital and Analog)
Output	12 V (Digital and Analog)	1.2, 3.3, and 5 V (depending on component)	-3.0, 3.0, and 3.3 V (depending on component)

	Oscillator	Buffer	FPGA
Input	3.3 V (Digital)	3.3 V Digital Square-wave signal (100 MHz)	● 3.3 V Digital Square-wave signal (100 MHz) (from Oscillator) ● 1.2 V Digital signal (from regulator) ● 3.3 V Digital signal (from regulator) ● Data from user (through JTAG/USB)

Output	3.3 V Digital Square-wave signal (100 MHz)	3.3 V Digital Square-wave signal (100 MHz) synchronized with the FPGA	8-bit (1 byte) output based on digital circuit created by FPGA

	DAC	Op-Amp	Low-pass filter
Input	● 8-bit (1 byte) output based on digital circuit created by FPGA ● 3.3 V (Digital) ● 3.3 V (Analog) ● 3.3 V Digital Square-wave signal (100 MHz) (from Oscillator)	In-phase (I wave) and quadrature (Q wave) signals which represent your output data in an analog way	Amplified in-phase (I wave) and quadrature (Q wave) signals to be used by the modulator
Output	In-phase (I wave) and quadrature (Q wave) signals which represent your output data in an analog way	Amplified in-phase (I wave) and quadrature (Q wave) signals to be used by the modulator	Amplified and filtered in-phase (I wave) and quadrature (Q wave) signals to be used by the modulator

	Modulator
Input	● Amplified and filtered in-phase (I wave) and quadrature (Q wave) signals ● Higher frequency carrier wave signal (f_{in})
Output	A choice of (f_{in} + I/Q waves) or (f_{in} - I/Q waves) which can be transmitted

IV. Summary

It is important to understand key signal concepts, such as a sinusoidal signal through both the time and spatial domains, as well as the relationship between the wavelength and frequency of a signal. This is significant when working with higher frequencies because of the potential noise created. Since the speed of light (in a vacuum) is constant, the wavelength will in contrast be small if the frequency is large. This can be seen from the equation below:

$$c = \lambda * f \qquad \text{(Equation 1)}$$

where λ is the wavelength of the signal, and f is the frequency of the signal.

Since the transmitted signal will be at a higher frequency, its wavelength will be small, thus having the full range of the signal witnessed in that short range, which can have a large-standing effect on the board. This effect shaped the way we designed our system and influenced the choices made during the selection of the different components.

As mentioned before we are going to use the Xilinx Spartan 6 (XC6SLX9-2TQG144C) due to its low cost, low power, high I/O performance, and relatively small PCB footprint. We decided to use the switch regulator to go from the power supply to a lower voltage (3.3v, 1.2v, 5v) due to its capability of handling a wider input voltage range compared to linear regulators, as well as its more efficient power consumption. In contrast, linear regulators tend to overheat when using a high input voltage. We use a linear regulator after the switch regulator so the component does not have to handle high voltages to reduce the risk of overheating. For the oscillator, we wanted a 100 MHz signal for our FPGA which is going to be used as our clock that can be easily synchronized with the FPGA. The DAC takes in 3.3v digital signals from the FPGA and converts it to analog signal (3.3v) so that the op-amp can amplify the signal so it can be used in our modulator. We use the low pass filters between the modulator and the op-amp so we can eliminate the high frequency noise that is created by the DAC and the op-amp.

KICAD SCHEMATICS (Work in Progress)

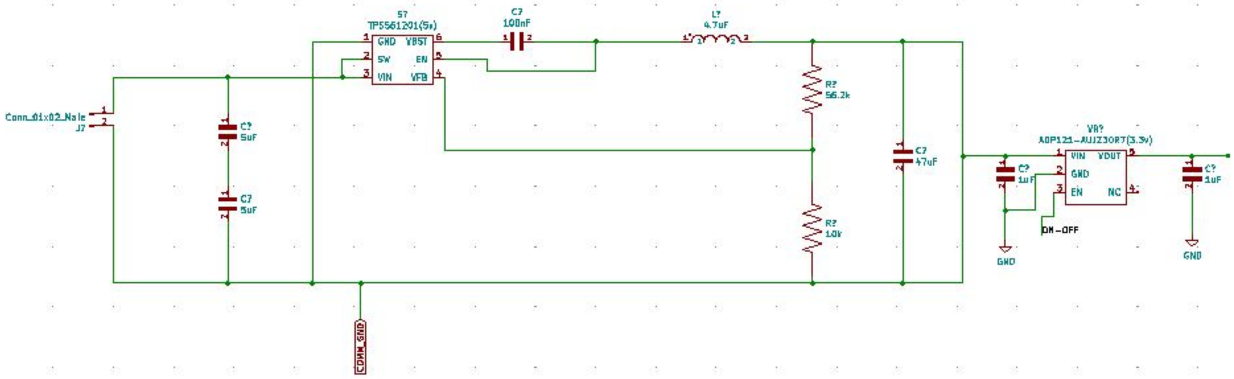

Figure 2: Switcher Regulator (5v) with Linear Regulator (3.3v)

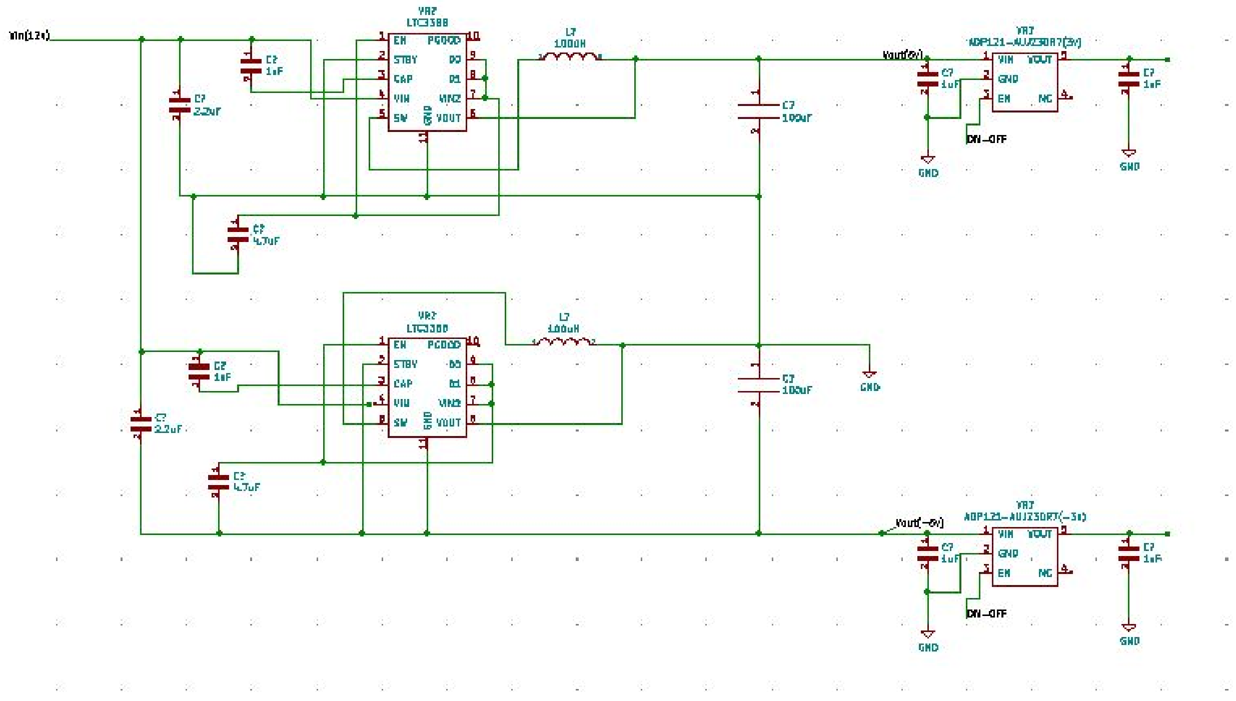

Figure 3: Split-Rail Switcher (5v,-5v) with Linear Regulators (3v,-3v)

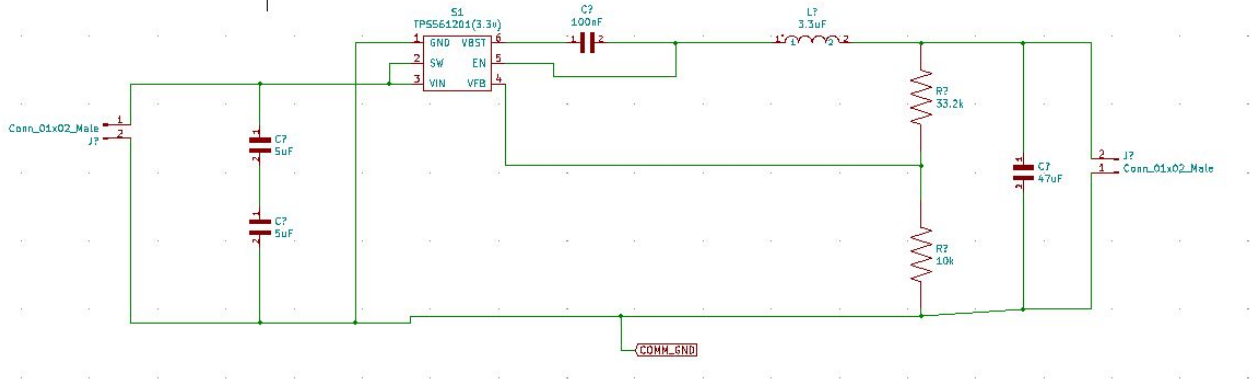

Figure 4: Switcher Regulator 3.3v

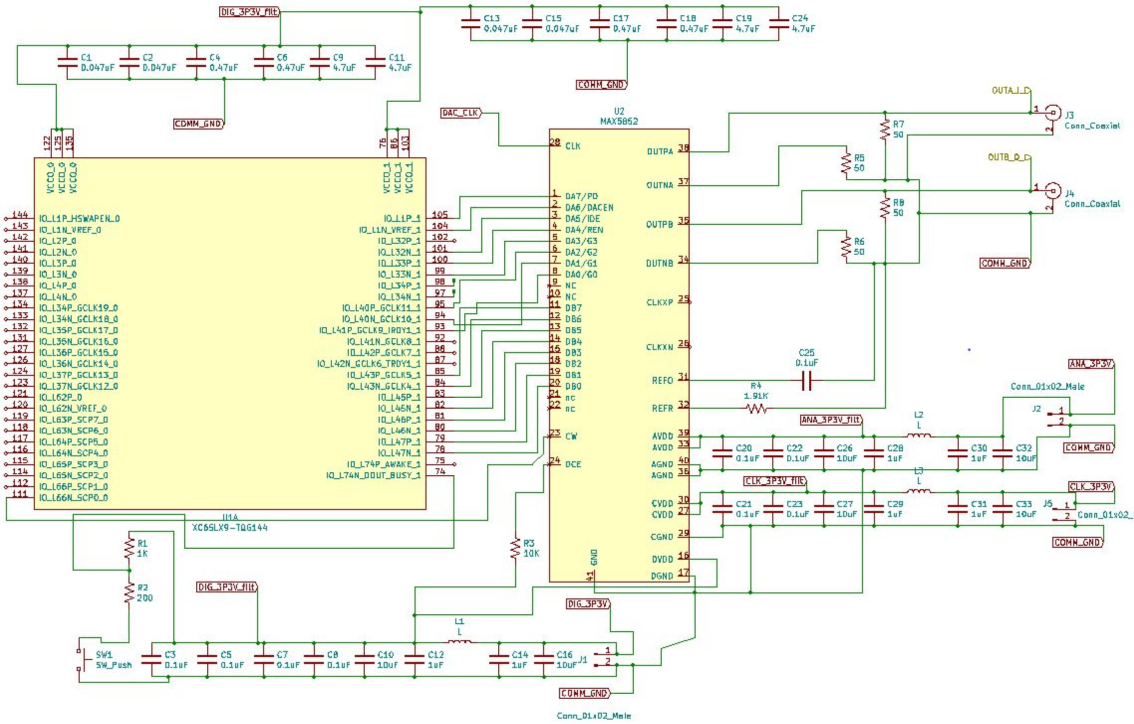

Figure 5: FPGA & DAC

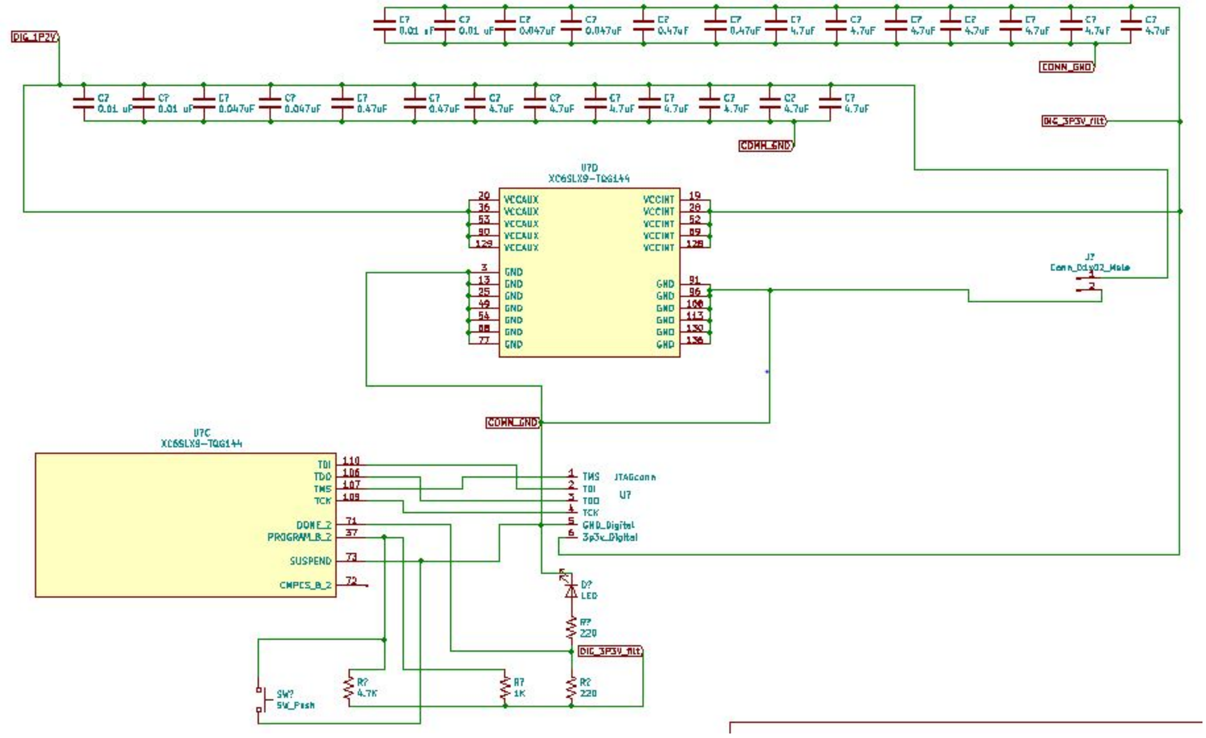

Figure 6: FPGA

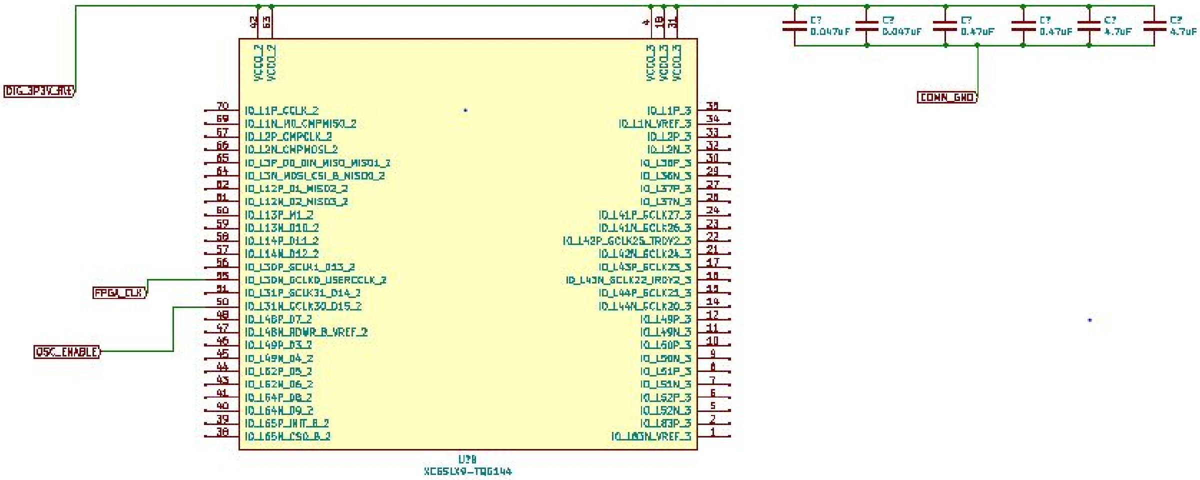

Figure 7: FPGA

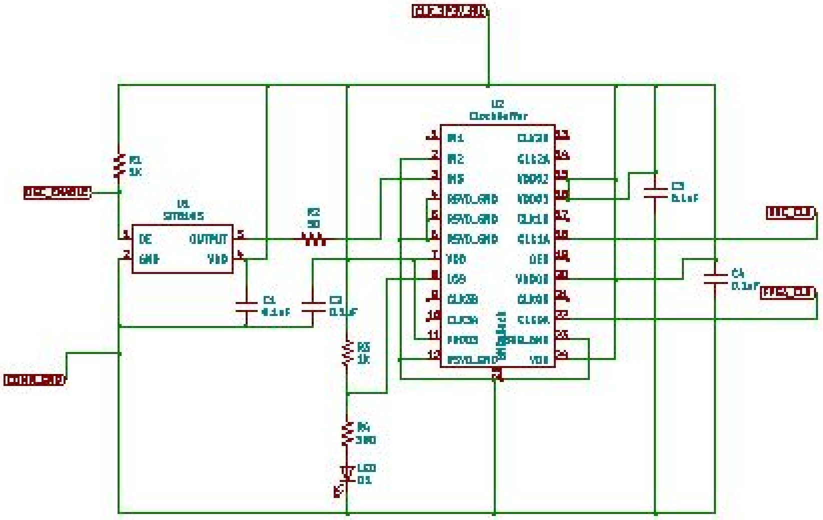

Figure 8: Oscillator & Buffer