The gross plan of this project is to synchronize systems where there is limited to no access to GPS or RF based signals. As a solution to this problem, we implemented a two-way time transfer and correction algorithm on an FPGA. Time stamps generated on the master and the slave site FPAGs are transferred between them and recorded. The slave site FPGA uses this data to compute the time error between the master and itself. By correcting its oscillator by this time error, we aim to synchronize the two systems.

The motivation behind selecting an FPGA involves but is not limited to the following factors,

1. An FPGA is one of the most agile reconfigurable devices and allows high reprogram ability.
2. It allows the design and use of elements such as a Custom Time to Digital Converters which allows timestamping with very high precision.
3. Moreover, since certain FPGA logic has the requirement of a clock signal, generation of such a signal and correcting it also becomes quite simple to accomplish when compared to a few other similar reconfigurable logic devices.

The milestones aimed to be achieved on the FPGA for this project are listed below,

1. For the FPGA to act as a controller orchestrating the flow of the Two-Way Time Transfer algorithm
2. For the FPGA to run the clock of interest and make corrections to the source of this clock (the oscillator)
3. To enable the system to also support retroreflectors
4. Transition coarse FPGA code for optical time transfer to contractor
5. The implementation of precision timing elements such as the Time to Digital Converter (TDC), that allows recording of incoming timing pulses with high precision. Test/characterize fine time transfer elements in isolation (i.e., outside of FSOTT system)
6. Transition fine time transfer code to contractor; Refine fine time transfer to achieve sub-ns precision time transfer

Items accomplished are,

1. Successfully designed a custom controller on the FPGA to act as the controller for the master and the slave-sites. This has enabled us to test the Corse Time-Transfer algorithm and achieve synchronization in the range of a few nanoseconds.
2. Since Sequential logic on the FPGA inherently requires a clock source, we have the Corse Time Stamp Generator on the FPGA and clock it with the clock of interest. The FPGA also houses the reprogrammable Oscillator that generates the clock source to the entire FPGA.
3. We currently are working on the incorporation of the precision timing elements into the system along with allowing the system to be compatible with retroreflectors.