# 6.111 Final Project Proposal: FPGA RFID Utility

*Miles Dai, Hannah Field*

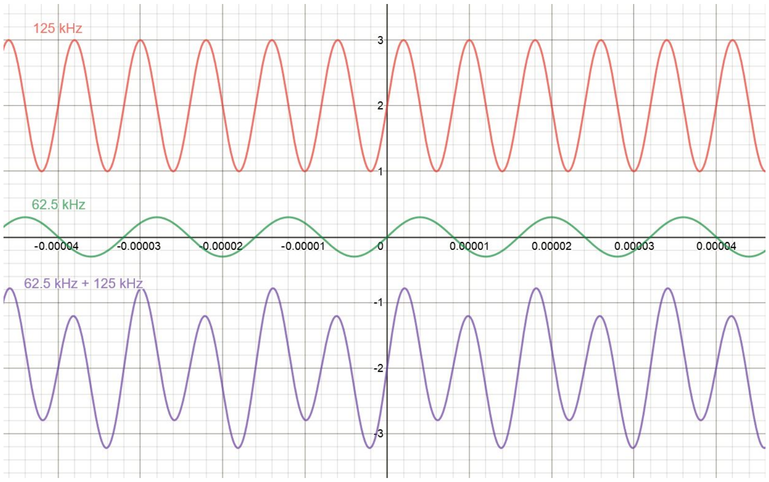
October, 2019

## Overview

In the fields of corporate and building security, contactless smartcards and proximity cards are the dominant form of access control. Indeed, Radio Frequency Identification (RFID) is the cornerstone of MIT's access control system. In recent years, non-contact forms of payment using Near-Field Communication (NFC) have also been growing in popularity. In our project, we would like to explore the security of this system by investigating the signals transmitted from these devices and attempting to replicate them.

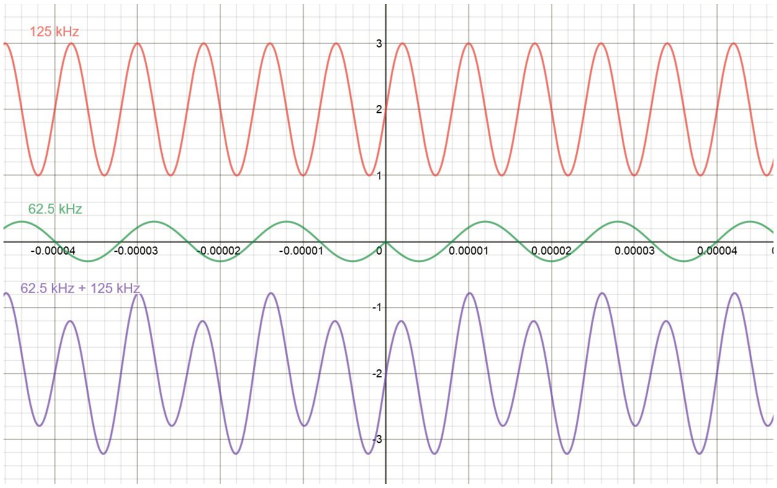
## Background

RFID is a subset of more general non-contact credential systems. In particular, MIT primarily uses passive RFID in the low frequency band, with card readers broadcasting 125kHz signals. Embedded within each ID card is a wire coil connected to an integrated circuit. This wire loop picks up the AC signal emitted by the card reader and rectifies it to provide power to the IC. The purpose of the IC is to effectively modulate the impedance across the ends of the coil. Because the coil in the card acts as the secondary of a transformer (with the card reader being the primary), the impedance changes in the card are reflected across the air gap and can be detected by the card reader.



*A 62.5kHz signal (green) superimposed on top of a 125kHz carrier (red) to form the purple waveform.*

These changes in impedance can be used to modulate the 125kHz signal. To transmit data, the IC on the card will superimpose a 62.5kHz sine wave on top, using the 125kHz signal as a carrier. This 62.5kHz wave transmits the stored ID number using binary phase-shift keying (BPSK).



*A phase-shifted signal (green) superimposed on the 125kHz carrier (red) to produce the purple waveform.*

RFID cards come in a variety of flavors. The onboard IC can be read-only, read-write, or write-once, read-many (WORM). Read-only cards have the ID number baked into the circuitry. Read-write cards allow for a card reader to edit the information on the card. WORM cards allow the end user to write to the card once, after which it becomes read-only.

# Implementation

The primary functionality of the project will consist of a reader and a broadcaster (spoofer).

## Card Reader

The reader will generate the 125kHz carrier signal which will be sent into a transmit coil through an amplifier, simulating a typical card reader. Meanwhile, the FPGA will read the signal emitted by the ID card through a second pickup coil. This signal will also be sent through an amplifier into the ADC on the Nexys 4 DDR board.

The protocol used by the MIT ID cards sends one bit every 16 cycles of the 62.5kHz signal. If a phase shift occurs between two consecutive bits, then the bit is flipped. Otherwise, the current bit is the same as the previous bit. The string of digits stored on the MIT ID cards consists of 30 zeros, 22 constant bits, 33 individualized bits, and 172 constant bits. Using a state machine, we can extract the individualized bits and store them within BRAM.

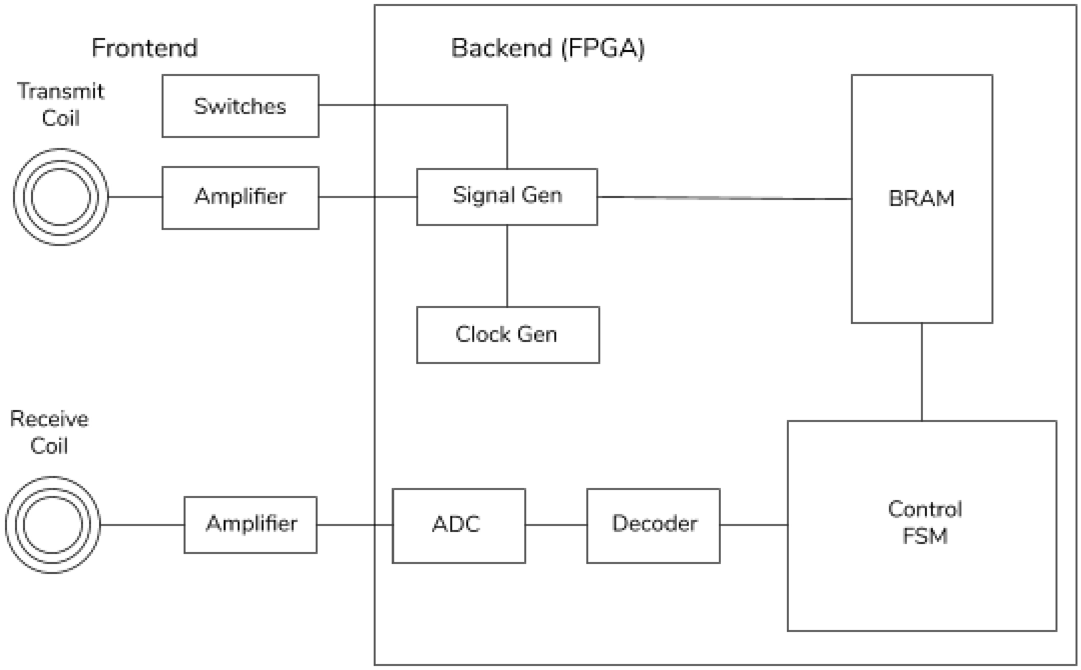
## Card Spoofer

Broadcasting the recorded bits can be accomplished through two possible methods: direct replay or generation of a BPSK signal. In the case of BPSK signal generation, the user will be able to specify the 33 personal bits via the switches on the FPGA.

The direct replay method uses hardware similar to that of the audio recording lab. The analog values of the incoming signal would simply be recorded into BRAM when reading an ID card. Because the protocol used in the MIT ID cards does not use any kind of temporal data (i.e. the card produces the exact same response every time it is excited by the 125kHz signal), these values can be replayed back directly into the card reader to spoof the original ID card.

The second method, generation of a BPSK signal, involves more complex hardware to recreate the signal produced by the card. However, it would only require storing the raw bits of a particular user’s ID card. This is tremendously more memory efficient and can also allow the system to spoof arbitrary ID numbers without actually having read the card. It also has the benefit of better noise immunity. As a result, we will most likely be pursuing this method of signal generation.

To actually produce the phase-shifted signal, it is more convenient not to generate the signal from scratch but rather to toggle a transistor connected to the ends of the coil. This can be done with the PMOD connectors on the board. Turning the transistor on effectively shorts the ends of the coil which will decrease the amplitude of the oscillations on the receive coil. This change will be reflected into the card reader which will interpret the resulting signal as that of an ID card.

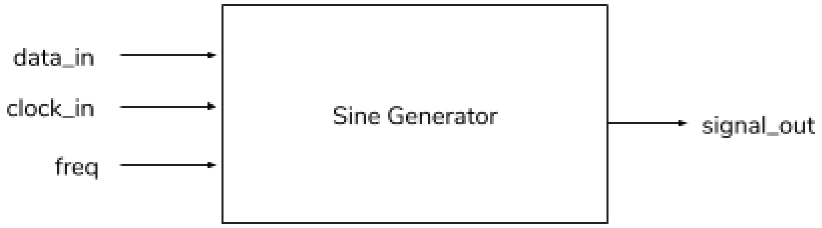


## Card Writing

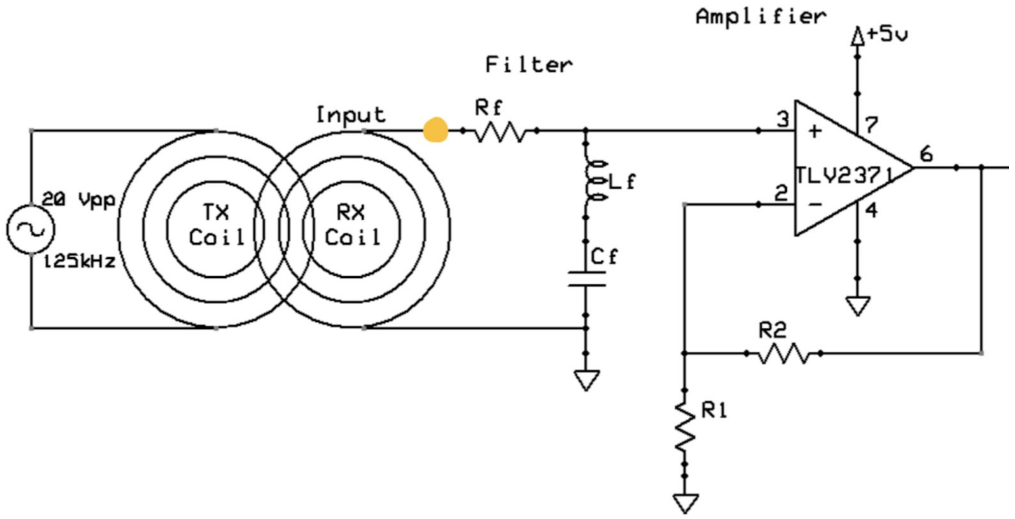
As an additional goal, we would like to investigate whether we can use our system to write the codes we have stored onto blank RFID cards. To this end, we can use the transmit coil and create a signal generator in the FPGA to generate the appropriate signals required to program the cards.

## Signal Generation

This section of the code is reminiscent of FPGA audio synthesizers. We use a sine-wave look up table and a register to keep track of an index into the table which effectively acts like a rotating phasor variable in order to generate sine waves. By changing how much we increment the index (i.e. rotate the phasor), we can change the frequency of the sine wave generated. Because we have precise control of the phasor register, we can also easily phase shift signals, allowing for easy production of phase-shift-keyed sine waves.



### Electrical System



The generation and absorption of the magnetic waves is accomplished by two hand-wound coils of wire. Data from the receive coil will first pass through a notch filter after which a TLV2371 op-amp with +15V and -15V rails will amplify the small signal going to be between 0 and 5V for optimal reading on the FPGA's ADC.

For the write and spoof functionality, the FPGA's outgoing DAC signal will pass through a second TLV2371 op-amp with +15V and -15V rails so that we may generate a signal with enough power to be received by MIT's card readers and also to power the RFID chip itself.

### Potential Hardware Limitations

The most likely limitation we will run into is the speed of the FPGA which directly impacts our ability to effectively synthesize waveforms.

## Required Materials

- Blank writable RFID cards

## Logistics

### Division of Work

- Analog Front-end (Hannah)
- 125kHz Signal Generation (Miles)
- BPSK Signal Generation (Miles)
- Receive Signal & State Machine (Hannah)

## Goals

- Read bits off MIT ID card
- Spoof the signal generated by the MIT ID card and open a card-reader controlled door

## Stretch Goals

- Make the entire system portable
- Write to blank RFID cards to duplicate cards stored in memory

## Timeline

- 11/3 - 11/9
    - Analog frontend
    - Read raw bits
- 11/10 - 11/16
    - Add ability to store ID numbers
    - Basic BPSK signal generation
- 11/17 - 11/23
    - Open card access door
- 11/24 - 11/30
    - Generate signals to write to cards
- 12/1 - 12/7
    - Work on portability and final touches